Εθνικό Μετσόβιο Πολυτεχνείο

Εργαστήριο Εμβιομηχανικής και Συστημικής Βιολογίας

ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΩΝ ΚΑΤΑΣΚΕΥΩΝ ΚΑΙ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημιουργία αλγορίθμου μηχανικής μάθησης για την υπολογιστική πρόβλεψη της έκκρισης πρωτεϊνών από το κύτταρο

Φοιτητής :ΦΩΤΗΣ ΧΡΗΣΤΟΣ

Επιβλέπων καθηγητής: Λεωνίδας Αλεξόπουλος
Επίκουρος Καθηγητής ΕΜΠ

ΑΘΗΝΑ 2018

Εθνικό Μετσόβιο Πολυτεχνείο

Εργαστήριο Εμβιομηχανικής και Συστημικής Βιολογίας

ΤΟΜΕΑΣ ΜΗΧΑΝΟΛΟΓΙΚΩΝ ΚΑΤΑΣΚΕΥΩΝ ΚΑΙ ΑΥΤΟΜΑΤΟΥ ΕΛΕΓΧΟΥ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημιουργία αλγορίθμου μηχανικής μάθησης για την υπολογιστική πρόβλεψη της έκκρισης πρωτεϊνών από το κύτταρο

Φοιτητής :ΦΩΤΗΣ ΧΡΗΣΤΟΣ

Επιβλέπων καθηγητής: Λεωνίδας Αλεξόπουλος
Επίκουρος Καθηγητής ΕΜΠ

Εγκρίθηκε τη 16η Φεβρουαρίου 2018 από την τριμελή επιτροπή:

…….	…….	…….
Αλεξόπουλος Λεωνίδας	Προβατίδης Χριστόφορος	Σπιτάς Βασίλειος
Επίκουρος Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ	Επίκουρος Καθηγητής ΕΜΠ

ΑΘΗΝΑ 2018

Περιεχόμενα.

1. Περίληψη

Η πληροφορία της έκκρισης μιας πρωτεΐνης είναι ύψιστης σημασίας τόσο για την ανακάλυψη νέων φαρμάκων όσο και για την ανακάλυψη νέων βιοδεικτών ασθένειας. Δυστυχώς όμως για ένα μεγάλο ποσοστό πρωτεΐνων δεν είναι γνωστό αν εκκρίνονται από το κύτταρο ή οχι (~20%). Από τη μοριακή βιολογία όμως είναι γνωστό ότι μια πρωτεΐνη μπορεί να εκκριθεί με δύο τρόπους. Είτε μέσω του συμβατικού μονοπατιού έκκρισης είτε μέσω μη συμβατικών μηχανισμών. Για την έκκριση μέσω του συμβατικού μονοπατιού καθοριστικό ρόλο παίζει η αλληλουχία των αμινοξέων της πρωτεΐνης και πιο συγκεκριμένα ή ύπαρξη χαρακτηριστικών περιοχών απο αμινοξέα που ονομάζονται σηματοδοτικές αλληλουχίες. Έτσι, αν μια πρωτεΐνη περιέχει σηματοδοτική αλληλουχία τότε εισέρχεται στο εκκριτικό μονοπάτι και μπορεί να καταλήξει είτε εκτός του κυττάρου, είτε εντός κάποιας μεμβράνης είτε σε κάποιο οργανίδιο του μονοπατιού εντός του κυττάρου. Είναι φανερό λοιπόν ότι για την πρόβλεψη της συμβατικής έκκρισης μια πρωτεΐνης η γνώση των σηματοδοτικών αλληλουχιών που περιέχει είναι πολύ σημαντική. Στα πλαίσια αυτής της διπλωματικής έρευνας το πρόβλημα της πρόβλεψης της έκκρισης μιας πρωτεΐνης αντιμετωπίστηκε χρησιμοποιώντας την πρόβλεψη διάφορων προγραμμάτων για την ύπαρξη σηματοδοτικών αλληλουχιών ως ενδιάμεση πληροφορία για την τελική πρόβλεψη της έκκρισης. Αφού εξακριβώθηκαν οι καλές επιδόσεις των προγραμμάτων των οποίων οι προβλέψεις χρησιμοποιήθηκαν ως δεδομένα, δημιουργήθηκε ένας αλγόριθμος μηχανικής μάθησης ο οποίος εκπαιδεύτηκε σε αυτά με σκοπό την πρόβλεψη της έκκρισης ή όχι της πρωτεΐνης. Η υλοποίηση του αλγορίθμου έγινε με τη μέθοδο random forest σε γλώσσα προγραμματισμού R και με τη χρήση του πακέτου caret

1. Abstract

Protein secretion plays a key role in both drug discovery and biomarker discovery by providing a distinguishing factor between useful and impractical drug targets and biomarkers. Unfortunately this information is not available for the whole proteome, with around 20% of human proteins still missing their secretion annotation. Molecular biology tells us that a protein can either be secreted through the secretory pathway or through unconventional mechanisms. In order for a protein to enter the secretory pathway, it must possess a characteristic domain of amino acids inside it's sequence, known as a signal sequence. Thus knowing whether or not a protein contains a signal sequence is crucial in predicting its secretion. There are several benchmarked machine learning methods to predict the presence of these domains given the sequence of a protein and our goal is to evaluate those methods and find their optimal combination to predict if a protein is secreted based on the predicted features. On that front, the output of several programs will be used as feature variables to train and test an ensemble model based on random forests for the final prediction of a protein's secretion.

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές |Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

2. Εισαγωγή

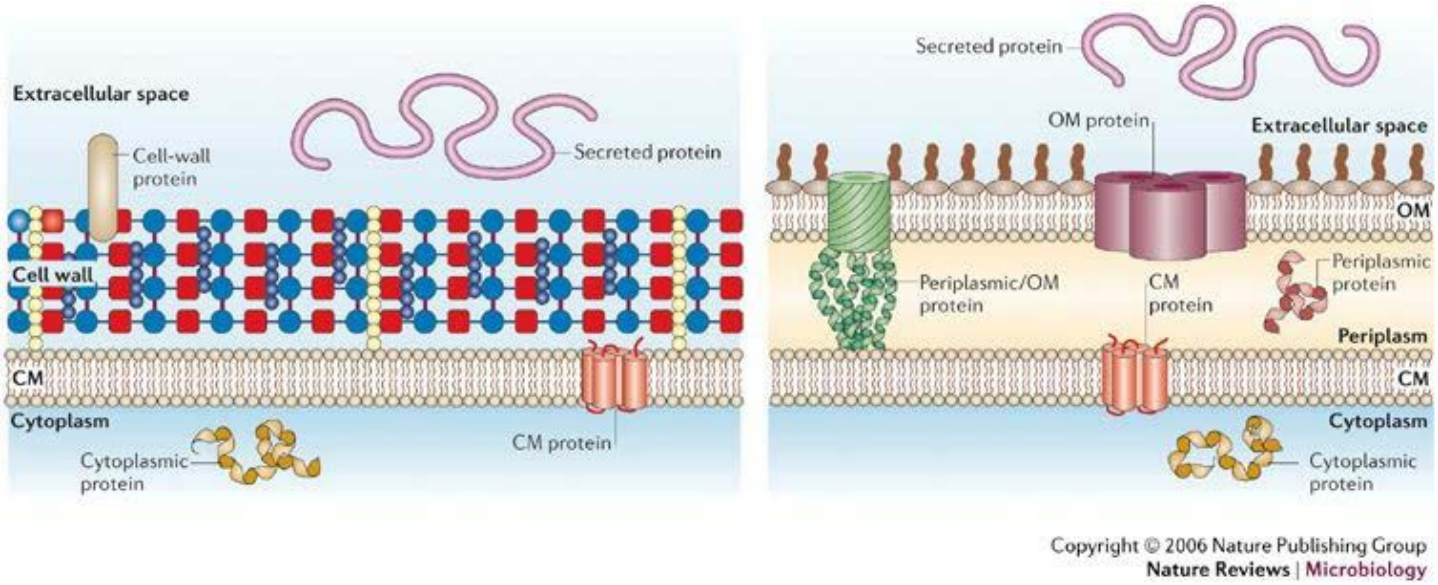
2.1. Σκοπός:

Σκοπός της παρούσας ερευνητικής εργασίας είναι η υπολογιστική πρόβλεψη της έκκρισης ή μη, μιας πρωτεΐνης από το κύτταρο, έχοντας ως δεδομένο την αλληλουχία των αμινοξέων που την αποτελούν. Για αυτό το λόγο, θα αναπτυχθεί ένα μοντέλο μηχανικής μάθησης, το οποίο θα εκπαιδευτεί σε πρωτεΐνες για τις οποίες είναι γνωστό εάν εκκρίνονται, με σκοπό την πρόβλεψη της έκκρισης σε νέα άγνωστα δείγματα.

2.2. Πρωτεΐνες και αμινοξέα:

Οι **πρωτεΐνες** αποτελούν τα πιο διαδεδομένα και πολυδιάστατα, τόσο στη μορφή όσο και στη λειτουργία τους, μακρομόρια. Αποτελούν είτε δομικά συστατικά του κυττάρου, είτε συνεργούν σε κάποια συγκεκριμένη κυτταρική λειτουργία. Πρόκειται για μεγάλα **σύνθετα βιομόρια** αποτελούμενα από **αμινοξέα**, τα οποία ενώνονται μεταξύ τους με πεπτιδικούς δεσμούς σχηματίζοντας μια γραμμική αλυσίδα, καλούμενη αλυσίδα πολυπεπτιδίων. Ο βιολογικός ρόλος τους καθορίζεται από την τρισδιάστατη δομή της, που με τη σειρά της εξαρτάται από την αλληλουχία των αμινοξέων που την αποτελούν. Η αλληλουχία αμινοξέων καθορίζεται από το γονίδιο του DNA που κωδικοποιεί την εκάστοτε πρωτεΐνη. Αξιοσημείωτο είναι πως στο ανθρώπινο DNA υπάρχουν 19613 γονίδια τα οποία κωδικοποιούν διαφορετικές πρωτεΐνες [1].

2.3. Κατηγοριοποίηση των πρωτεϊνων ανάλογα με την κυτταρική τους θέση



Εικόνα 1. Κατηγοριοποίηση των πρωτεϊνων ανάλογα με την κυτταρική τους θέση.

Στην εικόνα 1, παρουσιάζονται οι πιθανές γενικές κατηγορίες πρωτεϊνών ανάλογα με την τοποθεσία τους. Συνεπώς, οι γενικές κατηγορίες πρωτεϊνών είναι:

Α. Ενδοκυτταρικές πρωτεΐνες:

Οι ενδοκυτταρικές πρωτεΐνες εκτελούν το έργο τους ή εντός του κυττάρου, ή στο κυτταρόπλασμα ή στον πυρήνα ή σε κάποιο οργανίδιο εντός του κυττάρου (π.χ ενδοπλασματικό δίκτυο, μιτοχόνδρια).

Β. Εξωκυτταρικές πρωτεΐνες:

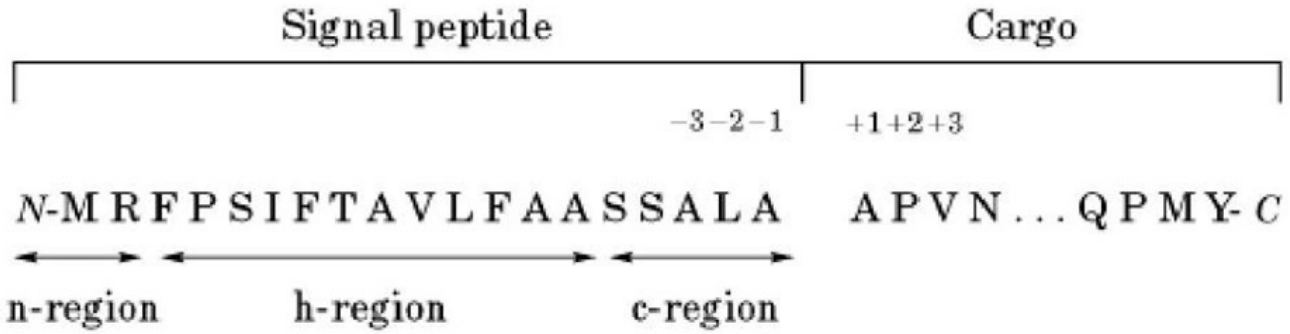
Οι εξωκυτταρικές πρωτεΐνες αντίθετα είναι οι πρωτεΐνες, που εκκρίνονται από το κύτταρο και χρησιμοποιούνται για την επικοινωνία των κυττάρων ή σαν δομικά υλικά της εξωκυττάριας μήτρας. Στον άνθρωπο παρόλο που όλα τα κύτταρα στο ανθρώπινο σώμα εκκρίνουν πρωτεΐνες, τα ενδοκρινικά κύτταρα και τα Β-λεμφοκύτταρα εξειδικεύονται στην έκκριση πρωτεϊνών. Οι εκκρινόμενες πρωτεΐνες έχουν πολύ μεγάλη σημασία στην ιατρική και τη βιολογία, αφενός λόγω του ότι αποτελούν το στόχο πολλών φαρμάκων και αφετέρου λόγω του ότι τα περισσότερα διαγνωστικά τεστ στοχεύουν τις πρωτεΐνες που μπορούν να βρεθούν είτε στο αίμα είτεστα ούρα. Οι εκκρινόμενες πρωτεΐνες, που βρίσκονται σε αφθονία, είναι τα παγκρεατικά ένζυμα (PRSS1, CELA3A, AMY2A) καθώς και άλλα πεπτικά ένζυμα (PGA3, PRR4, STATH).

Γ. Πρωτεΐνες μεμβράνης:
Οι πρωτεΐνες μεμβράνης αποτελούν μία απο τις πιο μεγάλες και πιο σημαντικές ομάδες πρωτεϊνών. Διαθέτουν ένα η περισσότερα τμήματα τους τοποθετημένα εντός της κυτταρικής ή κάποιας μεμβράνης ενός οργανιδίου (π.χ μιτοχόνδρια). Ο κύριος ρόλος τους ως μεταφορείς και δέκτες εξηγεί το λόγο για τον οποίο αποτελούν το στόχο του 59% όλων των εγκεκριμένων φαρμάκων [2].

Αξίζει να σημειωθεί πως η παρούσα ερευνητική εργασία θα εστιάσει στο αν μια πρωτεΐνη εκκρίνεται ή όχι από το κύτταρο, με τη θεώρηση πως οι εξωκυταρρικές πρωτεΐνες είναι οι πρωτεΐνες που εκκρίνονται, σε αντίθεση με τις ενδοκυταρρικές πρωτεΐνες και τις πρωτεΐνες μεμβράνης, οι οποίες θεωρείται πως δεν εκκρίνονται.

2.4. Σηματοδοτικές ακολουθίες:

Μέσα στην αλληλουχία αμινοξέων μιας πρωτεΐνης έχει ανακαλυφθεί ότι υπάρχουν ορισμένες χαρακτηριστικές περιοχές αμινοξέων, (λεγόμενες ως σηματοδοτικές αλληλουχίες), που καθορίζουν την κυτταρική τοποθεσία για την οποία προορίζεται η πρωτεΐνη, μετά την κωδικοποίηση της. Οι πιο σημαντικές από αυτές είναι:
Α. Σηματοδοτικά πεπτίδια.
Είναι μικρά πεπτίδια, συνήθως αποτελούμενα από 16-30 αμινοξέα και βρίσκονται στο N-άκρο της πρωτεΐνης. Παρόλο που δεν έχουν σταθερή σύνθεση αμινοξέων, όλα τα σηματοδοτικά πεπτίδια διέπονται από ορισμένους κανόνες. Αρχικά, ο πυρήνας του σηματοδοτικού πεπτιδίου αποτελείται από 5-16 υδροφοβικά αμινοξέα, τα οποία συνήθως δημιουργούν μία άλφα-έλικα* (περιοχή-h). Επιπλέον, τα σηματοδοτικά πεπτίδια συνήθως ξεκινούν με μια αλληλουχία θετικά φορτισμένων αμινοξέων κοντά στο N-άκρο της πρωτεΐνης (περιοχή-n). Τέλος, η περιοχή c περιέχει συνήθως τα αμινοξέα προλίνη και γλυκίνη [3] (εικόνα 2).



Εικόνα 2. Δομή σηματοδοτικού πεπτιδίου.

Β. Περιοχές μεμβράνης:

Οι περιοχές μεμβράνης αποτελούνται από υδροφοβικά αμινοξέα, έτσι ώστε να επιθυμούν να βρίσκονται μέσα στη διπλοστιβάδα λιπιδίων της μεμβράνης. Τα αμινοξέα των περιοχών μεμβράνης είναι συνήθως μη πολικά και 18-21 αμινοξέα είναι αρκετά ώστε να διασχίσουν την διπλοστιβάδα της εκάστοτε μεμβράνης [4].

** Με την ονομασία άλφα έλικα, ή έλικα άλφα, ή α-έλικα, (alpha helix, ή a-helix), φέρεται στη Βιολογία μια σπειροειδής αλυσίδα πολυπεπτιδίων, η οποία δημιουργεί μια ελικοειδή δομή σε πολλές πρωτεΐνες με 3,6 κατάλοιπα αμινοξέων ανά στροφή της έλικας. Οι διαδοχικές αυτές στροφές της άλφα έλικας συνδέονται με ασθενείς δεσμούς υδρογόνου και ως εκ τούτου, η δομή είναι περισσότερο σταθερή από μια μη σπειροειδή αλυσίδα πολυπεπτιδίων.*

2.5. Μηχανισμοί έκκρισης πρωτεϊνών:

Μια πρωτεΐνη η οποία διαθέτει σηματοδοτική αλληλουχία ακολουθεί το συμβατικό μονοπάτι έκκρισης. Το "μονοπάτι" αυτό αποτελείται από το ενδοπλασματικό δίκτυο, το σύμπλεγμα Golgi, τα λυσοσώματα και τα κυστίδια που ταξιδεύουν μεταξύ αυτών και της κυτταρικής μεμβράνης. Στο εκκριτικό μονοπάτι εισέρχονται όλες οι πρωτεΐνες που περιέχουν σηματοδοτική αλληλουχία, είτε αυτές προορίζονται για έκκριση από το κύτταρο είτε για τοποθέτηση μέσα σε κάποια μεμβράνη. Η διαδικασία που ακολουθείται είναι η εξής:

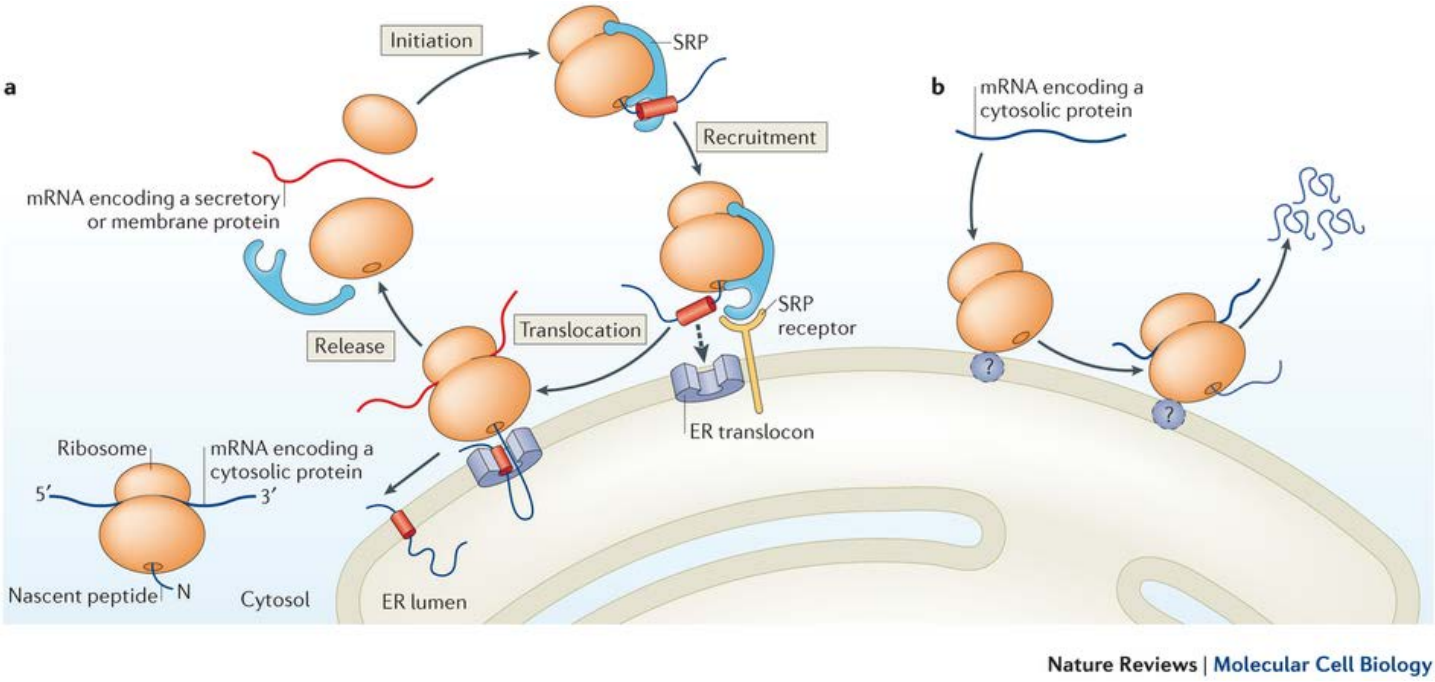
1. Όταν σ' ένα αγγελιοφόρο RNA (mRNA), που αρχίζει να μεταφράζεται από ένα ριβόσωμα στο κυτταρόπλασμα, εμπεριέχεται μία σηματοδοτική αλληλουχία,η μετάφραση σταματάει.

2. Η σηματοδοτική αλληλουχία αναγνωρίζεται από μια ριβονουκλεοπρωτεΐνη του ριβοσώματος (SRP) και το mRNA μεταφέρεται στο πρώτο στάδιο του εκκριτικού μονοπατιού, το ενδοπλασματικό δίκτυο. Η διαδικασία αυτή κατά την οποία η μετάφραση σταματάει όταν αναγνωριστεί κάποια σηματοδοτική αλληλουχία, ονομάζεται συμμεταφραστική μεταφορά (cotranslational translocation) (εικόνα 3).

4. Το ριβόσωμα μαζί με το μόριο SRP προσδένονται σε έναν υποδοχέα στη μεμβράνη του ενδοπλασματικού δικτύου και η μετάφραση της πρωτεΐνης συνεχίζεται.

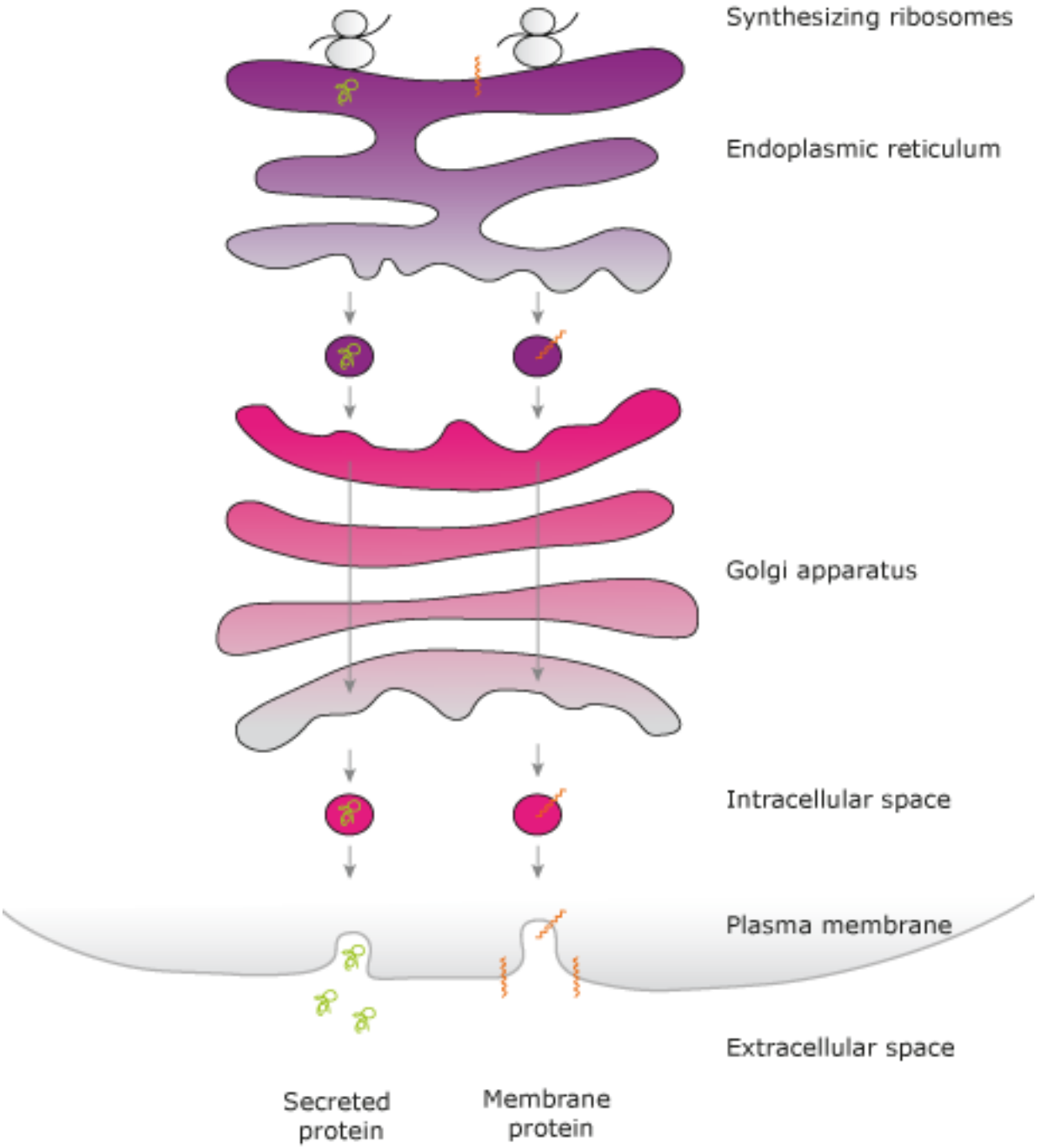
5. Η πρόσδεση γίνεται κοντά σε ένα κανάλι της πρωτεΐνης Sec61, έτσι ώστε η πρωτεΐνη να εισέλθει στην κοιλότητα του ενδοπλασματικού δικτύου καθώς μεταφράζεται.

Σε άλλες περιπτώσεις, η μεταφορά του RNA στο ενδοπλασματικό δίκτυο και στο εκκριτικό μονοπάτι γίνεται μετά τη μετάφραση και η διαδικασία αυτή ονομάζεται μετά-μεταφραστική μεταφορά (posttranslational translocation) [5].



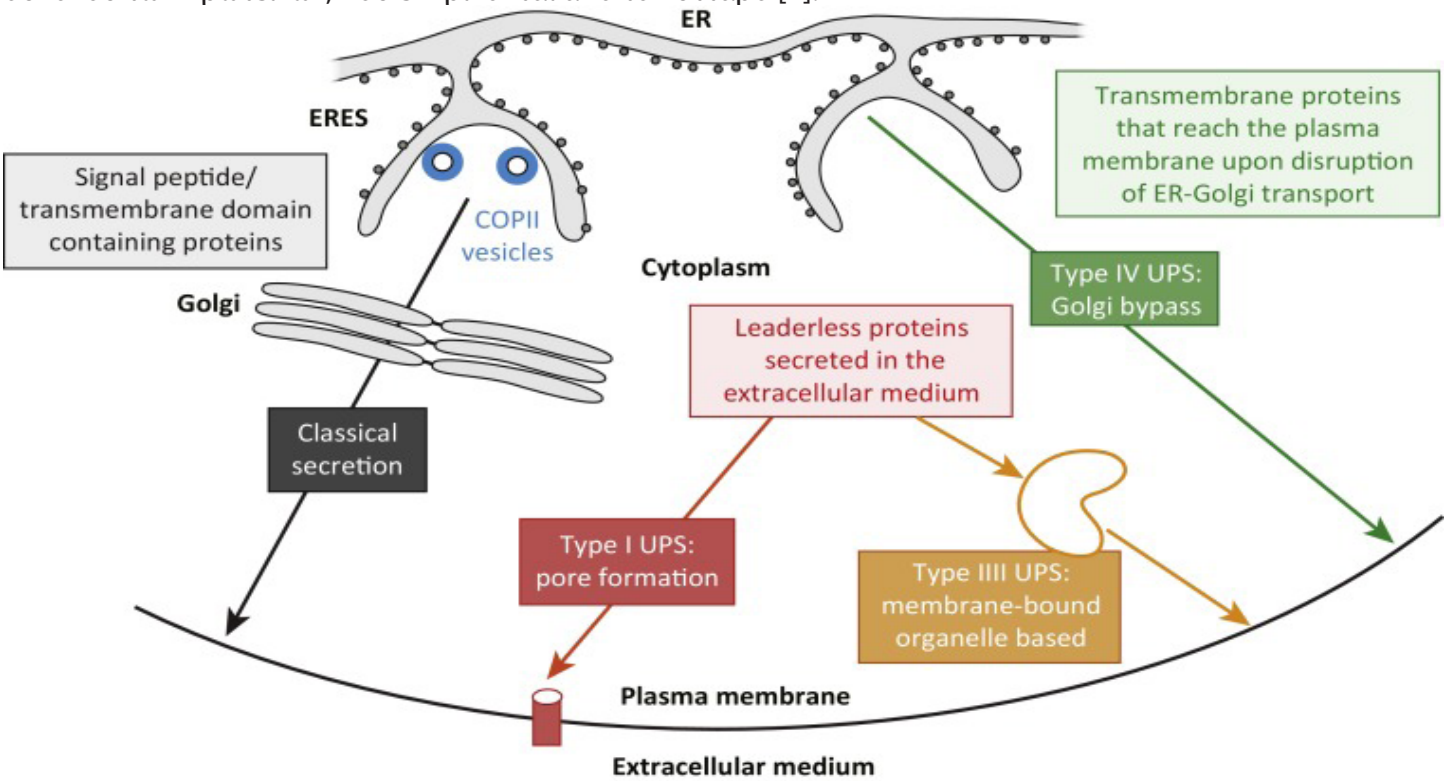
Εικόνα 3. Μεταφορά της πρωτεΐνης στο ενδοπλασματικό δίκτυο.

6. Οι περισσότερες πρωτεΐνες εγκαταλείπουν το ενδοπλασματικό δίκτυο μέσα σε μερικά λεπτά, μεταφερόμενες μέσα σε κυστίδια και κατευθύνονται προς το σύμπλεγμα Golgi και εν συνεχεία για έκκριση (εικόνα 4) [6]. Το σύμπλεγμα Golgi δεν είναι συνεχές αλλά αποτελείται από μια σειρά κυτταρικών οργανιδίων, που συντίθενται από μια δεσμίδα κυστιδίων – σάκων, τα οποία με τη σειρά τους συνδέονται με την κυτταροπλασματική μεμβράνη, γνωστοί ως ασκίδια ή δεξαμενές). Οι πρωτεΐνες δε μεταφέρονται μέσω κυστιδίων μέσα στα διάφορα τμήματα του συμπλέγματος Golgi εντούτοις το οργανίδιο στο οποίο βρίσκονται μετακινείται προς τα έξω και ωριμάζει με τη βοήθεια ενζύμων και εν συνεχεία μετατρέπεται στο επόμενο οργανίδιο.
7. Μετά το σύμπλεγμα Golgi, οι πρωτεΐνες μεταφέρονται μέσω κυστιδίων στον τελικό τους προορισμό, που είναι:
Α. Εξωκύττωση-ένωση με την κυτταρική μεμβράνη και έκκριση στο εξώκυττάριο περιβάλλον η παραμονή στην κυτταρική μεμβράνη.
Β. Εκκριτικά κυστίδια-εξωκύττωση μέσω κυστιδίων.
Γ. Λυσοσώματα, στα οποία αποδομούνται οι λανθασμένες πρωτεΐνες.



Εικόνα 4. Σχηματική αναπαράσταση της εκκριτικής πορείας

Παρά το γεγονός πως η πλειοψηφία των οι πρωτεϊνών που διαθέτουν σηματοδοτική αλληλουχία, εισέρχονται στο εκκριτικό μονοπάτι, εντούτοις δεν προορίζονται όλες για έκκριση ή για την κυτταρική μεμβράνη. Στο εκκριτικό μονοπάτι εισέρχονται κατά τη διαδικασία της μετάφρασής τους και οι πρωτεΐνες που επιτελούν τη λειτουργία τους μέσα στα οργανίδια του μονοπατιού (ενδοπλασματικό δίκτυο, σύμπλεγμα Golgi, κυστίδια).
Πέρα από το εκκριτικό μονοπάτι υπάρχουν και άλλοι μηχανισμοί μέσω των οποίων μια πρωτεΐνη μπορεί να εκκριθεί ή να καταλήξει στην κυτταρική μεμβράνη. Οι μηχανισμοί αυτοί ονομάζονται **μηχανισμοί μη συμβατικής έκκρισης** και περιλαμβάνουν (εικόνα 5):
Α. Παράκαμψη του συμπλέγματος Golgi και μεταφορά από το ενδοπλασματικό δίκτυο έξω από το κύτταρο (οι πρωτεΐνες αυτές περιέχουν σηματοδοτική αλληλουχία).
Β. Έκκριση πρωτεϊνών χωρίς σηματοδοτική αλληλουχία, με τη βοήθεια μεμβρανικών πόρων.
Γ. Έκκριση πρωτεϊνών χωρίς σηματοδοτική αλληλουχία, με τη βοήθεια οργανιδίων που προσδένονται στην κυτταρική μεμβράνη.
Οι πρωτεΐνες που ακολουθούν μηχανισμούς μη συμβατικής έκκρισης αναλογούν στο 10% περίπου του συνόλου των πρωτεϊνών, που εκκρίνονται από το κύτταρο [7].

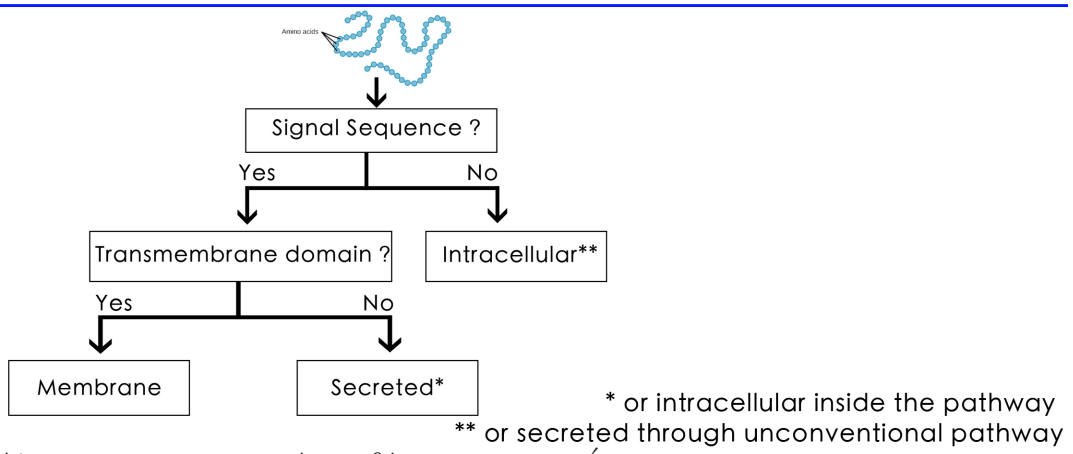


Εικόνα 5. Μηχανισμοί μη συμβατικής έκκρισης.

Trends in Cell Biology

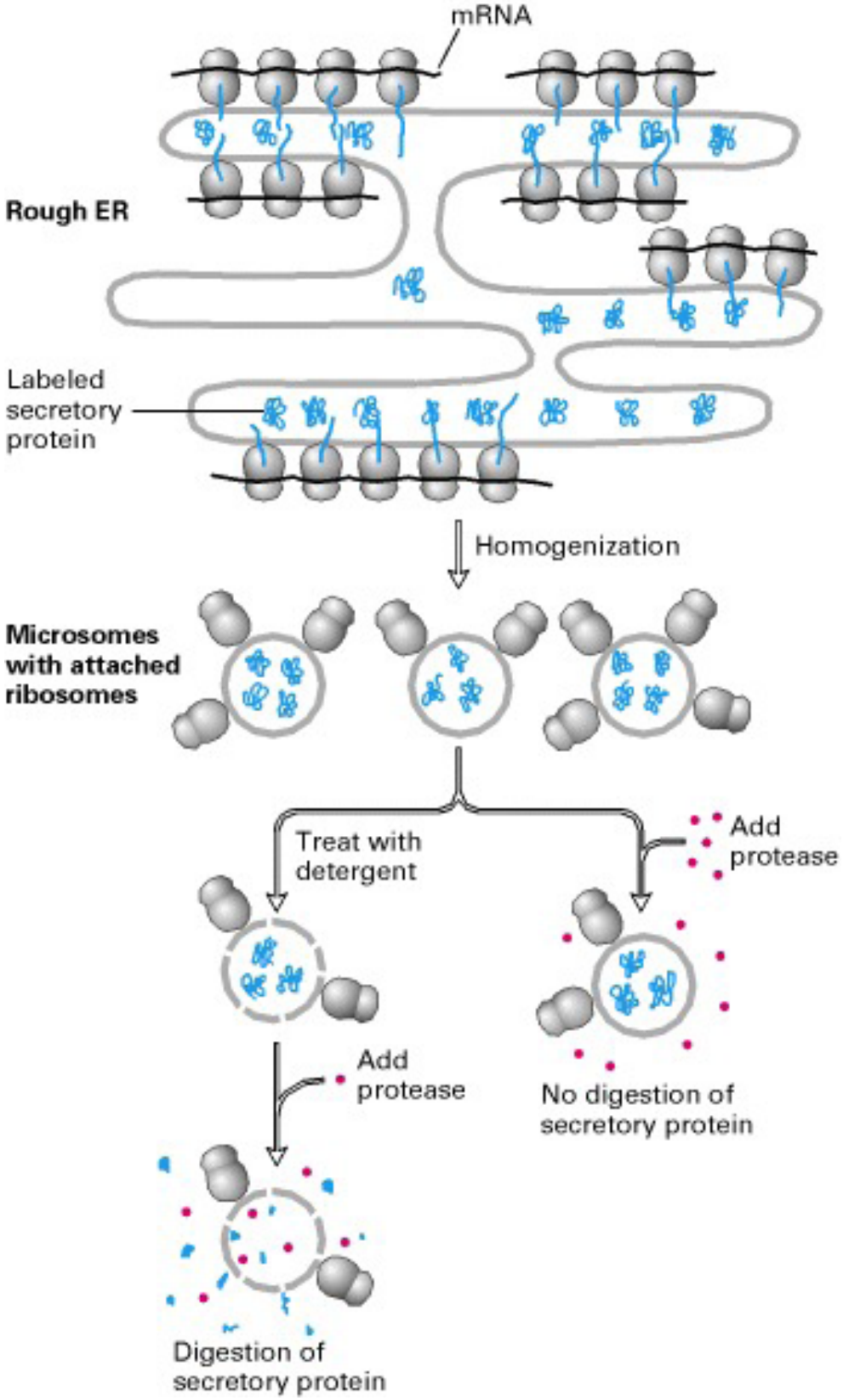
Ανακεφαλαιώνοντας η απόφαση του κυττάρου για την έκκριση μιας πρωτεΐνης μπορεί να περιγραφεί από το παρακάτω δέντρο απόφασης (εικόνα 6).

Biology



Εικόνα 6. Βιολογικό δέντρο απόφασης για την κυτταρική τοποθέτηση μιας πρωτεΐνης.

Η θέση των πρωτεϊνών μέσα στο κύτταρο έχει επαληθευτεί μέσω βιολογικών πειραμάτων. Τα βιολογικά αυτά πειράματα βασίζονται σε μία χαρακτηριστική ιδιότητα του ενδοπλασματικού δικτύου, το οποίο αφού καταστραφεί έχει τη δυνατότητα να ξανασυνδεθεί σε μικροσωμάτια. Απαραίτητα στοιχεία είναι επίσης η χρήση πρωτεάσης, που καταστρέφει τη συγκεκριμένη πρωτεΐνη που θέλουμε να εξακριβώσουμε την τοποθεσία της και η χρήση του αντίστοιχου αντισώματος για την μέτρησή της (εικόνα 7). Συνεπώς, η διαδικασία εξακρίβωσης εάν μια πρωτεΐνη εισέρχεται στο εκκριτικό μονοπάτι είναι η ακόλουθη :



Εικόνα 7. Σχηματική αναπαράσταση του βιολογικού πειράματος πρωτεϊνικής τοποθεσίας στο κύτταρο.

1. Αν η πρωτεΐνη μπορεί να μετρηθεί μετά τη χρήση πρωτεάσης αλλά δεν μπορεί να μετρηθεί μετά τη χρήση πρωτεάσης και διαλυτικού του ενδοπλασματικού δικτύου, τότε αυτή ανήκει στο εκκριτικό μονοπάτι, και είτε εκκρίνεται, είτε είναι πρωτεΐνη μεμβράνης, είτε εκτελεί τη λειτουργία της μέσα στο ίδιο το εκκριτικό μονοπάτι.
2. Στην αντίθετη περίπτωση, που μετά τη χρήση πρωτέασης, η πρωτεΐνη δε δύναται να μετρηθεί, τότε αυτή δεν ανήκει στο εκκριτικό μονοπάτι [8].

Από το σύνολο των πρωτεϊνών, που υπάρχουν στο ανθρώπινο σώμα γνωρίζουμε μόνο για μερικές από αυτές αν εκκρίνονται ή όχι με βάση την παραπάνω πειραματική διαδικασία. Ωστόσο, όπως αναλύθηκε και παραπάνω, από τους μηχανισμούς έκκρισης του κυττάρου η ύπαρξη ή μη μιας σηματοδοτικής αλληλουχίας στην πρωτεΐνη είναι καλός προβλεπτής για να προσδιορίσουμε την τοποθεσία της πρωτεΐνης μέσα στο κύτταρο. Δυστυχώς όμως δεν είναι γνωστό για όλες τις πρωτεΐνες αν αυτές περιέχουν ή όχι σηματοδοτική αλληλουχία. Για αυτό το λόγο, έχουν αναπτυχθεί προγράμματα τεχνητής νοημοσύνης που δέχονται σαν είσοδο την αλληλουχία αμινοξέων μιας πρωτεΐνης και προβλέπουν την ύπαρξη ή όχι σηματοδοτικών αλληλουχιών σε αυτές.

2.6. Πρόβλεψη σηματοδοτικών αλληλουχιών

Κοινό χαρακτηριστικό όλων των προγραμμάτων πρόβλεψης ύπαρξης σηματοδοτικής αλληλουχίας ή τοποθεσίας της πρωτεΐνης είναι ότι δέχονται ως είσοδο την αλληλουχία αμινοξέων και έχουν εκπαιδευτεί να αναγνωρίζουν τις σηματοδοτικές αλληλουχίες με βάση τις πρωτεΐνες, για τις οποίες πειραματικά έχει επιιβεβαιωθεί ότι περιέχουν σηματοδοτική αλληλουχία. Υπάρχει πληθώρα προγραμμάτων πρόβλεψης σηματοδοτικών πεπτιδίων και περιοχών μεμβράνης, τα οποία διατίθενται για τοπική εγκατάσταση. Στην παρούσα έρευνα θα εστιάσουμε στα πιο γνωστά προγράμματα τα οποία έχει αποδειχτεί ότι έχουν τις καλύτερες επιδόσεις. Για μια εκτεταμένη λίστα προγραμμάτων πρόβλεψης ο αναγνώστης παραπέμπεται εδώ:
(https://www.expasy.org/resources/search/keywords:subcellular%20location) [9].

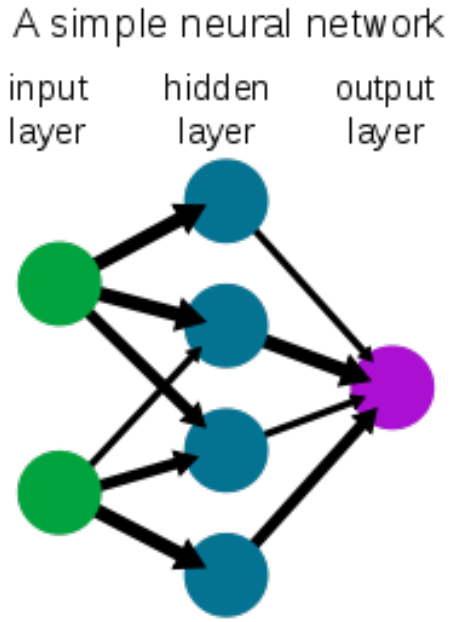
2.7. Προγράμματα πρόβλεψης σηματοδοτικών πεπτιδίων

SignalP

Το πρόγραμμα SignalP προβλέπει την ύπαρξη και τη θέση του σηματοδοτικού πεπτιδίου σε μία πρωτεΐνη. Το πρόγραμμα είναι ένα τεχνητό νευρωνικό δίκτυο, που σαν είσοδο δέχεται την αλληλουχία αμινοξέων της πρωτεΐνης και προβλέπει για κάθε ένα αμινοξύ αν αυτό αποτελεί μέρος του σηματοδοτικού πεπτιδίου.
Το νευρωνικό δίκτυο είναι ένα δίκτυο από απλούς υπολογιστικούς κόμβους (νευρώνες, νευρώνια), διασυνδεδεμένους μεταξύ τους. Είναι εμπνευσμένο από το Κεντρικό Νευρικό Σύστημα (ΚΝΣ), το οποίο και προσπαθεί να προσομοιώσει.
Οι νευρώνες είναι τα δομικά στοιχεία του δικτύου. Κάθε τέτοιος κόμβος δέχεται ένα σύνολο αριθμητικών εισόδων από διαφορετικές πηγές (είτε από άλλους νευρώνες, είτε από το περιβάλλον), επιτελεί έναν υπολογισμό με βάση αυτές τις εισόδους και παράγει μία έξοδο. Η εν λόγω έξοδος είτε κατευθύνεται στο περιβάλλον, είτε τροφοδοτείται ως είσοδος σε άλλους νευρώνες του δικτύου. Υπάρχουν τρεις τύποι νευρώνων: οι νευρώνες εισόδου, οι νευρώνες εξόδου και οι υπολογιστικοί νευρώνες ή κρυμμένοι νευρώνες. Οι νευρώνες εισόδου δεν επιτελούν κανέναν υπολογισμό, μεσολαβούν απλώς ανάμεσα στις περιβαλλοντικές εισόδους του δικτύου και στους υπολογιστικούς νευρώνες. Οι νευρώνες εξόδου διοχετεύουν στο περιβάλλον τις τελικές αριθμητικές εξόδους του δικτύου. Οι υπολογιστικοί νευρώνες πολλαπλασιάζουν κάθε είσοδό τους με το αντίστοιχο συναπτικό βάρος και υπολογίζουν το ολικό άθροισμα των γινομένων. Το άθροισμα αυτό τροφοδοτείται ως όρισμα στη συνάρτηση ενεργοποίησης, την οποία υλοποιεί εσωτερικά κάθε κόμβος. Η τιμή που λαμβάνει η συνάρτηση για το εν λόγω όρισμα είναι και η έξοδος του νευρώνα για τις τρέχουσες εισόδους και βάρη.
Εάν xki είναι η I-οστή έισοδος του k νευρώνα, wki: το I-στο συνοπτικό βάρος του k νευρώνα και φ() η συνάρτηση ενεργοποίησης του νευρωνικού δικτύου, τότε η έξοδος yk του k νευρώνα δίνεται από την εξίσωση:

$$y_k = \phi\left(\sum_{i=0}^{N} x_{ki} w_{ki}\right) \qquad (1)$$

Στον k-οστό νευρώνα υπάρχει ένα συνοπτικό βάρος wk0 με ιδιαίτερη σημασία, το οποίο καλείται πόλωση η κατώφλι. Η τιμή της εισόδου του είναι πάντα η μονάδα.
Εάν το συνολικό άθροισμα από τις υπόλοιπες εισόδους του νευρώνα είναι μεγαλύτερο από την τιμή αυτή, τότε ο νευρώνας ενεργοποιείται. Εάν είναι μικρότερο, τότε ο νευρώνας παραμένει ανενεργός. Η ιδέα προέκυψε από τα βιολογικά νευρικά κύτταρα.
Όπως είναι φανερό, οι αριθμοί οι οποίοι συναποτελούν το διάνυσμα εισόδου (κάθε στοιχείο του διανύσματος τροφοδοτείται κατά τη λειτουργία του δικτύου σε έναν νευρώνα εισόδου) αλλά και οι αριθμοί οι οποίοι συναποτελούν το διάνυσμα εξόδου (κάθε στοιχείο του οποίου εμφανίζεται, μετά το πέρας του ολικού υπολογισμού, σε έναν νευρώνα εξόδου), περιγράφουν χαρακτηριστικά του προς επίλυση προβλήματος. Συνήθως αυτό που μας ενδιαφέρει είναι το δίκτυο να απεικονίζει με ορθό τρόπο διανύσματα εισόδου σε κατάλληλα διανύσματα εξόδου, το πρόβλημα δηλαδή είναι η υλοποίηση μίας συνάρτησης πολλαπλών μεταβλητών, κατά κανόνα περίπλοκης και με άγνωστο ακριβή τύπο. Τέτοιες απεικονίσεις έχουν εφαρμογή σε ποικιλία τομέων της επιστήμης και της τεχνολογίας, αφού λειτουργούν ως αριθμητικά μοντέλα για πολλά διαφορετικά ζητήματα. Το ίδιο δίκτυο μπορεί να υλοποιήσει άπειρες διαφορετικές απεικονίσεις, μία για κάθε διαφορετική επιλογή συνόλου συναπτικών βαρών (εικόνα 8).
Το κύριο χαρακτηριστικό των νευρωνικών δικτύων είναι η εγγενής ικανότητα μάθησης. Ως μάθηση, μπορεί να οριστεί η σταδιακή βελτίωση της ικανότητας του δικτύου να επιλύει κάποιο πρόβλημα (π.χ. η σταδιακή προσέγγιση μίας συνάρτησης). Η μάθηση επιτυγχάνεται μέσω της εκπαίδευσης, μίας επαναληπτικής διαδικασίας σταδιακής προσαρμογής των παραμέτρων του δικτύου (συνήθως των βαρών και της πόλωσής του) σε τιμές κατάλληλες ώστε να επιλύεται με επαρκή επιτυχία το προς εξέταση πρόβλημα. Αφού ένα δίκτυο εκπαιδευτεί, οι παράμετροί του συνήθως «παγώνουν» στις κατάλληλες τιμές και από εκεί κι έπειτα είναι σε λειτουργική κατάσταση. Το ζητούμενο είναι το λειτουργικό δίκτυο να χαρακτηρίζεται από μία ικανότητα γενίκευσης: αυτό σημαίνει πως δίνει ορθές εξόδους για εισόδους καινοφανείς και διαφορετικές από αυτές με τις οποίες εκπαιδεύτηκε [10].
Σαν είσοδος στο πρόγραμμα SignalP συγκεκριμένα δίνονται τα 70 πρώτα αμινοξέα μιας πρωτεΐνης καθώς έχει αποδειχθεί ότι απο εκεί και μετά η πρωτεΐνη δεν μπορεί να έχει σηματοδοτικό πεπτίδιο. Η έξοδος είναι η πιθανότητα το εκάστοτε αμινοξύ να ανήκει σε σηματοδοτικό πεπτίδιο. Για κάθε αμινοξύ, η είσοδος στο νευρωνικό δίκτυο δίνεται μέ ένα κινούμενο παράθυρο 13 θέσεων, έτσι το νευρωνικό δίκτυο κάθε φορά κάνει την πρόβλεψη για το συγκεκριμένο αμινοξύ, έχοντας πληροφοριές για 6 αμινοξέα πιο πίσω και 6 αμινοξέα πιο μπροστά στην ακολουθία. Το πρόβλημα αντιμετωπίζεται σαν πρόβλημα ταξινόμησης τεχνητής μάθησης και το νευρωνικό δίκτυο εκπαιδεύεται σε δεδομένα πρωτεϊνών ευκαρυωτικών κυττάρων για τα οποία γνωρίζουμε αν περιέχουν σηματοδοτικά πεπτίδια [11]. Οι παράμετροι του νευρωνικού δικτύου σε κάθε κύκλο επανάληψης ρυθμίζονται ώστε να ελαχιστοποιείται η συνάρτηση σφάλματος:

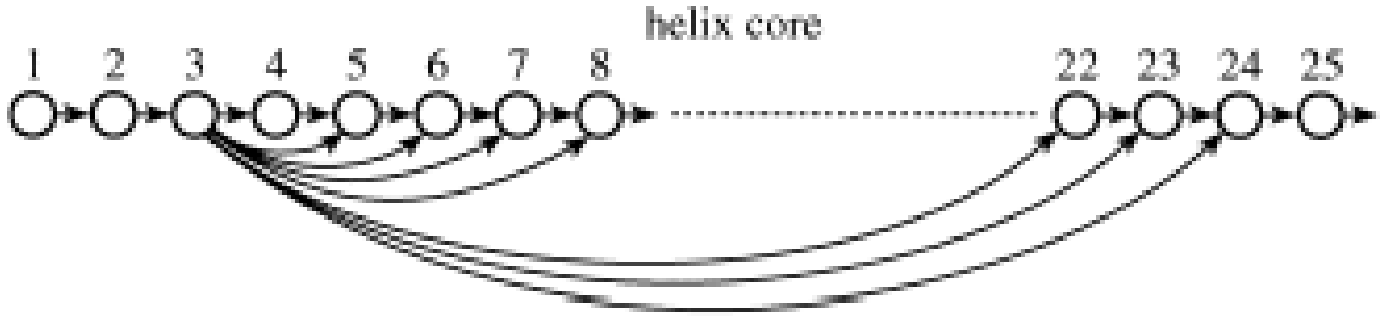


Εικόνα 8. Σχηματική αναπαράσταση ενός απλού νευρωνικού δικτύου.

2.8. Προγράμματα πρόβλεψης περιοχών μεμβράνης

TMHMM

Το πρόγραμμα TMHMM προβλέπει την ύπαρξη περιοχών μεμβράνης στην αλληλουχία μιας πρωτεΐνης καθώς και την τοπολογία της διαμεμβρανικής πρωτεΐνης με τη χρήση ενός hidden markov model (HMM). Η αρχιτεκτονική του μοντέλου φαίνεται στην εικόνα 9. Στην εικόνα 9α, κάθε κουτί αντιστοιχεί σε ένα υπομοντέλο σχεδιασμένο να μοντελοποιεί μια συγκεκριμένη περιοχή της διαμεμβρανικής πρωτεΐνης. Κάθε υπομοντέλο περιέχει πολλές HMM καταστάσεις για να μοντελοποιήσει το διαφορετικό μήκος των διάφορων περιοχών. Τα βέλη δείχνουν τον τρόπο με τον οποίο γίνονται οι μεταβάσεις μεταξύ των μοντέλων, έτσι ώστε να υπακούουν στους κανόνες των διαμεμβρανικών πρωτεϊνών. Όπως φαίνεται στην εικόνα, το υπομοντέλο globular επιτρέπει μετάβαση μόνο στον εαυτό του και στο υπομοντέλο loop στην πλευρά εντός του κυττάρου. Η τοπολογία κοντά στην μεμβράνη μοντελοποιείται στα υπομοντέλα loop και cap που φαίνεται στην εικόνα 9b. Τα υπομοντέλα loop είναι 3 διαφορετικά αλλά όλα έχουν 20 καταστάσεις, οι οποίες έχουν την ίδια κατανομή αμινοξέων. Το υπομοντέλο cap μοντελοποιεί την αρχή της μεμβρανικής περιοχής μιας πρωτεΐνης με μήκος 5 αμινοξέα. Το υπομοντέλο για τον πυρήνα της διαμεμβρανικής περιοχής έχει 25 πιθανές καταστάσεις, με τη δυνατότητα μετάβασης από μια κατάσταση σε όλες τις πιθανές καταστάσεις. Το μήκος του πυρήνα επιτρέπεται να είναι μεταβλητό απο 5 έως 25 αμινοξέα. Το πραγματικό μήκος έτσι εξαρτάται από τις πιθανότητες μετάβασεις από την μία κατάσταση στην άλλη, μέσα στον πυρήνα της διαμεμβρανικής περιοχής. Οι παράμετροι του HMM μοντέλου είναι οι πιθανότητες εμφάνισης του κάθε αμινοξέως στην παρατηρήσιμη κατάσταση και οι πιθανότητες μετάβασης από μια κατάσταση σε άλλη, που καθορίζουν το μήκος της κάθε περιοχής. Οι παράμετροι για τα μοντέλα προσεγγίστηκαν από ένα σετ 160 πρωτεϊνών, στις οποίες οι διαμεμβρανικές περιοχές έχουν εξακριβωθεί πειραματικά. Η πρόβλεψη των μεμβρανικών περιοχών, δεδομένου του HMM γίνεται βρίσκοντας την πιο πιθανή τοπολογία (αυτή που ταιριάζει στο μοντέλο).
Στο πρόγραμμα TMHMM δέχεται σαν είσοδο την αλληλουχία αμινοξέων ολόκληρης της πρωτεΐνης και επιστρέφει τον αριθμό των διαμεμβρανικών περιοχών, τη θέση τους και την τοπολογία της πρωτεΐνης μεμβράνης [12].



Εικόνα 9. Αρχιτεκτονική του μοντέλου TMHMM

2.9. Προγράμματα πρόβλεψης σηματοδοτικών πεπτιδίων και μεμβρανικών περιοχών

Phobius

Το πρόγραμμα Phobius δέχεται σαν είσοδο την αλληλουχία αμινοξέων μια πρωτεΐνης και προβλέπει την ύπαρξη σηματοδοτικών πεπτιδίων και μεμβρανικών περιοχών. Η μέθοδος είναι παρόμοια με το πρόγραμμα TMHMM καθώς και στο Phobius χρησιμοποιούνται υπομοντέλα HMM για να μοντελοποιήσουν τις περιοχές μεμβράνης και τα σηματοδοτικά πεπτίδια. Η αρχιτεκτονική του μοντέλου είναι παρόμοια με αυτή του TMHMM και φαίνεται στην εικόνα 10.

Με τη συνδυασμένη πρόβλεψη σηματοδοτικών πεπτιδίων και περιοχών μεμβράνης βελτιώνεται η ακρίβεια πρόβλεψης καθώς αντιμετωπίζεται το πρόβλημα της λάθος κατηγοριοποίησης ενός σηματοδοτικού πεπτιδίου, ως περιοχή μεμβράνης [13].

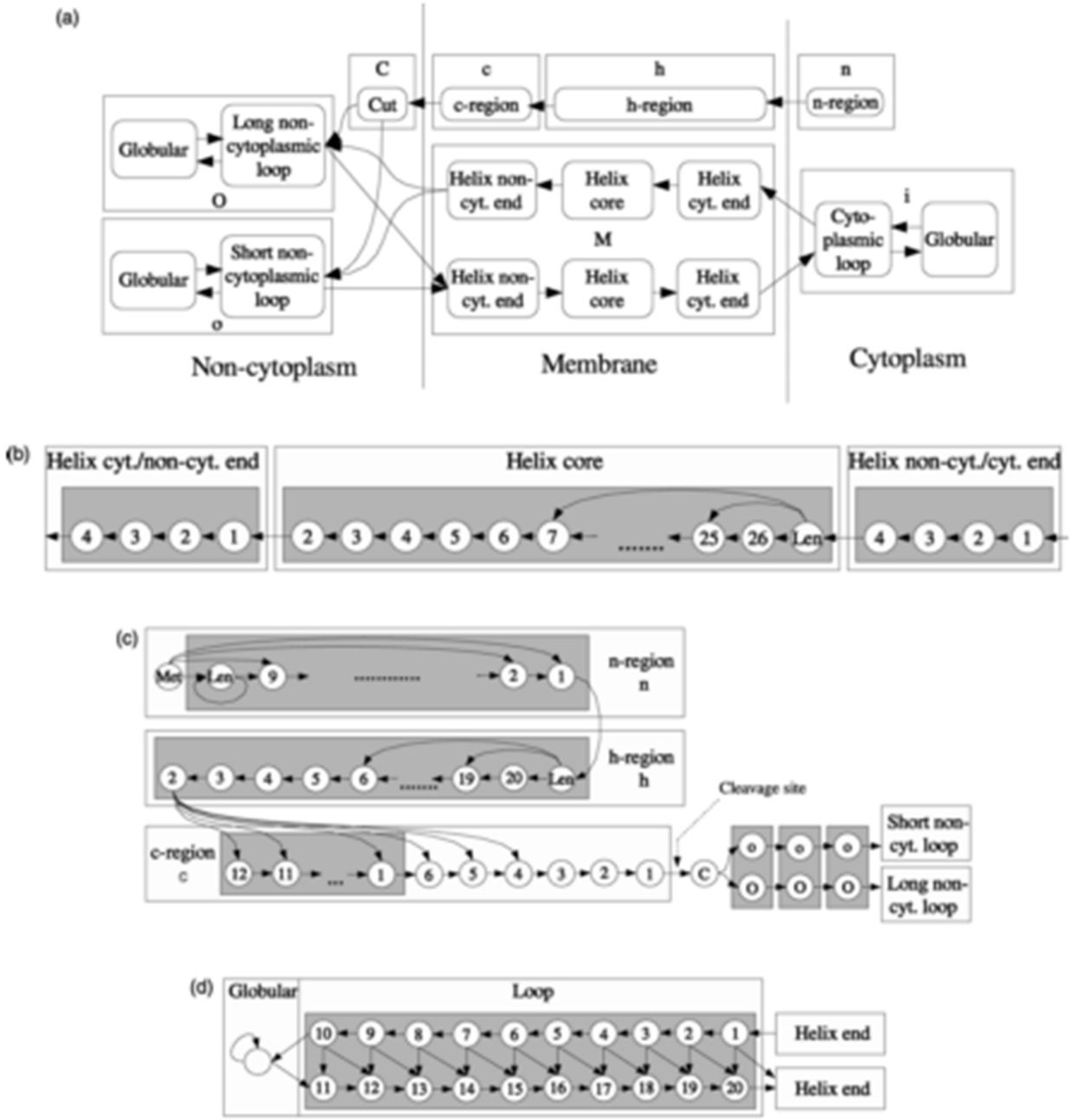


Figure 1. The layout of the HMM model. States within grayed boxes have tied emission probabilities. (a) Overview of the model. (b) The TM helix submodel. (c) The signal peptide submodel. (d) The cytoplasmic and short non-cytoplasmic loop submodels.

Εικονα 10. Αρχιτεκτονική του μοντέλου HMM του Phobius.

TargetP

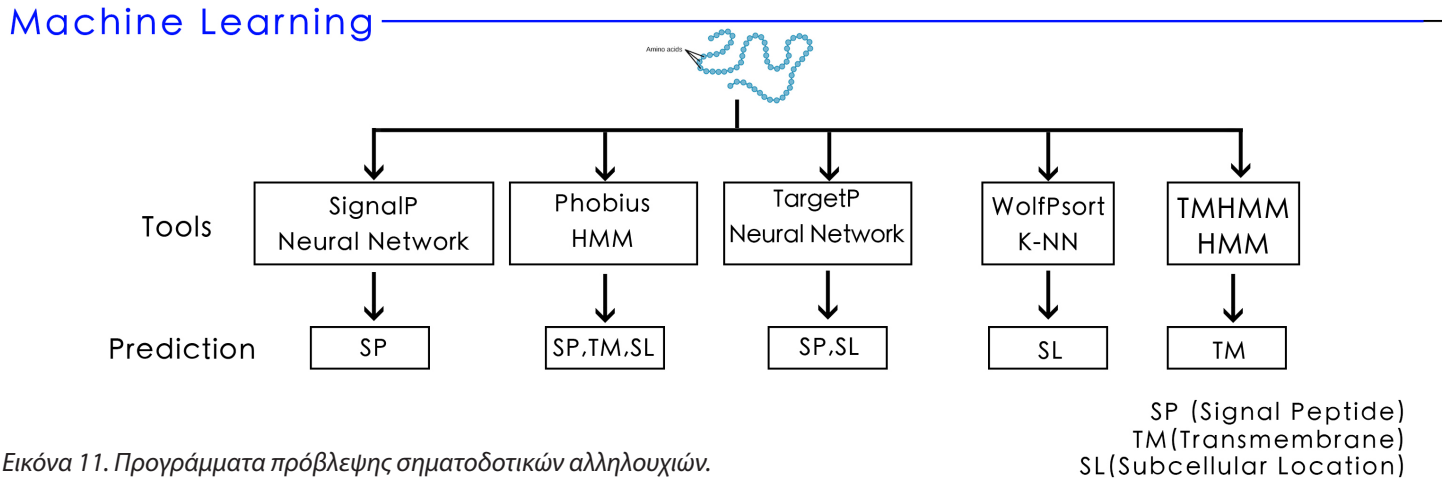
Το πρόγραμμα TargetP συνδυάζει τα προγράμματα SignalP και TMHMM για να προβλέψει τα σηματοδοτικά πεπτίδια και τις περιοχές μεμβράνης μίας πρωτεΐνης. Το χαρακτηριστικό του είναι ότι αντίθετα με τα άλλα προγράμματα, έχει εκπαιδευτεί σε δεδομένα, τα οποία περιέχουν και πρωτεΐνες βακτηρίων, μυκήτων και φυτών. Για αυτό το λόγο, οι επιδόσεις του είναι χειρότερες όταν οι είσοδοι προς πρόβλεψη είναι μόνο ανθρώπινες πρωτεΐνες. Η βασική αρχή λειτουργίας του είναι ίδια με τα προγράμματα SignalP και TMHMM για την πρόβλεψη των σηματοδοτικών πεπτιδίων και περιοχών μεμβράνης αντιστοίχως [14].

2.10. Προγράμματα πρόβλεψης τοποθεσίας πρωτεΐνης

WolfPSORT

Το πρόγραμμα WolfPSORT προβλέπει την τελική τοποθεσία μιας πρωτεΐνης. Δέχεται σαν είσοδο την αλληλουχία αμινοξέων της πρωτεΐνης και υπολογίζει κάποιες χαρακτηριστικές μεταβλητές, όπως την ύπαρξη ή μη σηματοδοτικού πεπτιδίου, την ύπαρξη περιοχής μεμβράνης, τον αριθμό των αμινοξέων κ.ά. Στη συνέχεια, με βάση αυτές τις χαρακτηριστικές μεταβλητές χρησιμοποιεί τον αλγόριθμο k-nearest neighbors για να κατηγοριοποιήσει τις πρωτεΐνες στις αντίστοιχες τοποθεσίες. Ο k-NN είναι ένας απλός αλγόριθμος machine learning, που για να κατηγοριοποιήσει ένα νέο δείγμα κοιτάει τα k πιο κοντινά δείγματα στο χώρο των χαρακτηριστικών μεταβλητών, με βάση κάποια απόσταση και μετά από ψηφοφορία μεταξύ αυτών κατηγοριοποιεί το δείγμα στην ίδια ομάδα με την πλειοψηφία.
Έξοδος του προγράμματος είναι η τελική τοποθεσία της πρωτεΐνης καθώς και το ποσοστό της πλειοψηφίας κατά την ψηφοφορία μεταξύ των k-γειτόνων [15].

Στην παρούσα έρευνα στόχος μας είναι να προβλέψουμε αν μια πρωτεΐνη εκκρίνεται από το κύτταρο ή οχι. Για αυτό το λόγο θα δημιουργήσουμε έναν αλγόριθμο τεχνητής νοημοσύνης που θα δέχεται σαν είσοδο την αλληλουχία αμινοξέων μιας πρωτεΐνης και θα χρησιμοποιεί σαν χαρακτηριστικές μεταβλητές την ύπαρξη ή οχι σηματοδοτικών αλληλουχιών μέσα στη πρωτεΐνη. Οι χαρακτηριστικές μεταβλητές, που θα χρησιμοποιηθούν για την πρόβλεψη, θα είναι η έξοδος των προγραμμάτων πρόβλεψης που έχουν αναφερθεί (εικόνα 11).



Εικόνα 11. Προγράμματα πρόβλεψης σηματοδοτικών αλληλουχιών.

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές | Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

3. Δεδομένα πρωτεϊνών

Για τη δημιουργία του προγράμματος πρόβλεψης της έκκρισης ή μη μίας πρωτεΐνης χρησιμοποιήθηκε η βάση δεδομένων Uniprot (http://www.uniprot.org/). Εν συνεχεία, μέσω του Uniprot λήφθηκε το σετ δεδομένων, που περιλαμβάνει 20183 πρωτεΐνες, η εγκυρότητα των οποίων έχει ελεγχθεί από τους κριτές της βάσης δεδομένων.

3.1. Μετασχηματισμός των δεδομένων

Σε πρώτο στάδιο, από το σύνολο των 20183 πρωτεϊνών αφαιρέθηκαν όσες δεν έχουν καμμία πληροφορία τόσο για την τοποθεσία τους, όσο και για την ύπαρξη ή μη κάποιας σηματοδοτικής αλληλουχίας. Σε δεύτερο στάδιο αφαιρέθηκαν ορισμένες διπλές είσοδοι στα δεδομένα και στο τελικό στάδιο απέμειναν 16233 πρωτεΐνες. Πρόκειται για πρωτεΐνες, για τις οποίες γνωρίζουμε εάν εκκρίνονται ή όχι καθώς και εάν έχουν σηματοδοτική αλληλουχία. Ωστόσο, η μορφή στην οποία βρίσκεται αυτή η πληροφορία δε δύναται να χρησιμοποιηθεί στην εκπαίδευση ενός προγράμματος μάθησης μηχανής. Η αρχική μορφή της στήλης subcellular location στα δεδομένα μας φαίνεται παρακάτω για παράδειγμα για την πρωτεΐνη Q68CP4.

"SUBCELLULAR LOCATION: Lysosome membrane {ECO:0000269|PubMed:16960811, ECO:0000269|-PubMed:17033958, ECO:0000269|PubMed:17897319}; Multi-pass membrane protein {ECO:0000269|-PubMed:16960811, ECO:0000269|PubMed:17033958, ECO:0000269|PubMed:17897319}. Note=Colocalizes with the lysosomal marker LAMP2. The signal peptide is not cleaved upon translocation into the endoplasmic reticulum; the precursor is probably targeted to the lysosomes via the adapter protein complex-mediated pathway that involves tyrosine- and/or dileucine-based conserved amino acid motifs in the last C-terminus 16-amino acid domain."

Ως εκ τούτου, για κάθε πρωτεΐνη διατίθεται μία παράγραφος, που παρέχει πληροφορία για τις θέσεις στις οποίες έχει βρεθεί η εκάστοτε πρωτεΐνη και για τις έρευνες που τις έχουν επαληθεύσει (PubMed ID). Προκειμένου να εξαχθεί από αυτή την παράγραφο η έκκριση ή μη της πρωτεΐνης ακολουθήθηκαν τα ακόλουθα βήματα:
1. Αφαίρεση των επιπλέον σημειώσεων. (Notes).
2. Διαχωρισμός της παραγράφου σε μικρότερες, έτσι ώστε καθεμία να περιέχει μία εξακριβωμένη θέση, για την εκάστοτε πρωτεΐνη και την αντίστοιχη έρευνα.
3. Δημιουργία ενός λεξιλογίου, το οποίο ψάχνει στην παράγραφο για λέξεις, όπως "secreted", "extracellular", οι οποίες εάν βρεθούν, υποδηλώνεται η έκκριση της πρωτεΐνης.
Στη συνέχεια αυτής της διαδικασίας εμφανίστηκαν κάποιες πρωτεΐνες, οι οποίες είχαν αντικρουόμενες πληροφορίες για το εάν εκκρίνονται ή όχι από διαφορετικές έρευνες. Αποφασίστηκε η αφαίρεση αυτών των πρωτεινών καθώς είναι απαραίτητο οι κατηγορίες των δειγμάτων να είναι καθαρές (εκκρίνεται-δεν εκκρίνεται), προκειμένου να εκπαιδευτεί ένας αλγόριθμος κατηγοριοποίησης/ταξινόμησης.
Εν τέλει, **ο αριθμός των πρωτεΐνών που παρέμειναν είναι 12274**.

2.3. Διερεύνηση δεδομένων

Στο μετασχηματισμένο σετ των δεδομένων είναι σημαντικό να εξεταστεί η συσχέτιση, που είναι γνωστή από τη μοριακή βιολογία ότι υπάρχει, μεταξύ της ύπαρξης σηματοδοτικής αλληλουχίας στην πρωτεΐνη και την έκκρισή της από το κύτταρο.

Table 1: Secreted ~ Signal Peptide

	No SP	SP
Not secreted	10436	1199
Secreted	22	1067

Πίνακας 1. Συσχέτιση έκκρισης με την ύπαρξη σηματοδοτικού πεπτιδίου.

Στον πίνακα 1 παρατηρείται πως η μη ύπαρξη σηματοδοτικού πεπτιδίου έχει ισχυρή συσχέτιση με τη μη έκκριση της πρωτεΐνης, εν αντιθέσει με την ύπαρξη σηματοδοτικού πεπτιδίου, που δεν έχει ισχυρή συσχέτιση με την έκκριση της πρωτεΐνης. Συνεπώς υπάρχουν 1199 πρωτεΐνες, οι οποίες έχουν σηματοδοτικό πεπτίδιο αλλά δεν εκκρίνονται από το κύτταρο. Αυτό συμβαίνει γιατί αυτές οι πρωτεΐνες πέρα από σηματοδοτικό πεπτίδιο, περιέχουν και περιοχές μεμβράνης με αποτέλεσμα να μην εκκρίνονται. Επιπλέον, παρατηρείται ότι το σετ των δεδομένων περιλαμβάνει 11635 πρωτεΐνες που δεν εκκρίνονται και 1089 που εκκρίνονται. Αξίζει να σημειωθεί πως υπάρχει ισχυρή ανισσοροπία στα δεδομένα, γεγονός που θα πρέπει να αντιμετωπιστεί κατά την εκπαίδευση και έλεγχο του αλγορίθμου ταξινόμησης.

Table 2: Secreted ~ TM domain

	No TM	TM
Not secreted	7626	4009
Secreted	1087	2

Πίνακας 2. Συσχέτιση έκκρισης με ύπαρξη περιοχής μεμβράνης.

Στον πίνακα 2 παρατηρείται πως υπάρχει ισχυρή συσχέτιση μεταξύ των διαμεμβρανικών περιοχών και της μη έκκρισης της πρωτεΐνης. Εν αντιθέσει με τη μη ύπαρξη περιοχής μεμβράνης, η οποία δε συσχετίζεται με την έκκριση ή τη μη έκκριση της πρωτεΐνης.

Table 3: Secreted ~ Signal Peptide + TM domain

	TM + NO SP	NO TM + NO SP	TM + SP	NO TM + SP
Not secreted	2987	7449	1027	172
Secreted	1	21	1	1066

Πίνακας 3. Συσχέτιση σηματοδοτικής αλληλουχίας με έκκριση.

Στον πίνακα 3 παρατηρείται πως όταν συνδυάζονται οι πληροφορίες για τα 2 είδη σηματοδοτικών αλληλουχιών, υπάρχει ισχυρή συσχέτιση μεταξύ όλων των χαρακτηριστικών μεταβλητών και των δύο κατηγοριών (έκκριση-μη έκκριση). Στην περίπτωση που η πρωτεΐνη περιέχει σηματοδοτικό πεπτίδιο μόνο (μεταβλητή NO TM + SP) υπάρχουν 172 πρωτεΐνες που δεν εκκρίνονται. Οι πρωτεΐνες αυτές θεωρητικά εισέρχονται στο εκκριτικό μονοπάτι αφού έχουν σηματοδοτικό πεπτίδιο αλλά ανήκουν στην κατηγορία των πρωτεΐνων που παραμένουν και εκτελούν τη λειτουργία τους σε κάποιο οργανίδιο μέσα στο μονοπάτι. Επιπλέον, υπάρχουν 21 πρωτεΐνες που δεν έχουν καθόλου σηματοδοτική αλληλουχία (μεταβλητή NO TM + NO SP) αλλά εκκρίνονται θεωρητικά μέσω μηχανισμών μη συμβατικής έκκρισης.

Table 4: Location ~ Signal Peptide + TM domain

	TM + NO SP	NO TM + NO SP	TM + SP	NO TM + SP
Intracellular	2	7208	1	162
Membrane	2985	241	1026	10
Secreted	1	21	1	1066

Πίνακας 4. Συσχέτιση σηματοδοτικής αλληλουχίας με τοποθεσία της πρωτεΐνης.

Στον πίνακα 4 παρουσιάζονται οι ίδιες χαρακτηριστικές μεταβλητές και το πως αυτή τη φορά συσχετίζονται με την τοποθεσία της πρωτεΐνης. Παρατηρείται πως οι πρωτεΐνες που έχουν σηματοδοτικό πεπτίδιο και δεν εκκρίνονται, στην πλειοψηφία τους δεν είναι πρωτεΐνες μεμβράνης αλλά ενδοκυτταρικές μέσα στο εκκριτικό μονοπάτι.

Εν συνεχεία, το σύνολο αυτό των μετασχηματισμένων δεδομένων θα χρησιμοποιηθεί αρχικά για τον έλεγχο των επιδόσεων των προγραμμάτων πρόβλεψης σηματοδοτικών αλληλουχιών και για την εκπαίδευση και έλεγχο του αλγορίθμου ταξινόμησης.

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές | Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

4. Εκτίμηση των επιδόσεων των προγραμμάτων πρόβλεψης

Στην ενότητα αυτή θα εξεταστεί **η επίδοση των προγραμμάτων πρόβλεψης,** οι έξοδοι των οποίων θα χρησιμοποιηθούν ως χαρακτηριστικές μεταβλητές για την πρόβλεψη της έκκρισης ή μη μιας πρωτεΐνης. Συνεπώς, η καλή τους επίδοση είναι απαραίτητη προκειμένου η έξοδος να μπορεί να χρησιμοποιηθεί ως προβλεπτής για την έκκριση ή μη της πρωτεΐνης.

4.1. Δείκτες επίδοσης μάθησης μηχανής

Η πρόβλεψη των σηματοδοτικών αλληλουχιών και της τοποθεσίας της πρωτεΐνης είναι πρόβλημα κατηγοριοποίησης/ταξινόμησης με δύο πιθανές κατηγορίες-κλάσεις στην κάθε περίπτωση.
Συγκεκριμένα:

1) **για τις σηματοδοτικές αλληλουχίες οι κατηγορίες-κλάσεις είναι περιέχει-δεν περιέχει σηματοδοτική αλληλουχία** και
2) **για την έκκριση πρωτεΐνων είναι εκκρίνεται-δεν εκκρίνεται (TRUE-FALSE)**.

Για το χαρακτηρισμό των επιδόσεων αλγορίθμων κατηγοριοποίησης εξαιρετική σημασία έχει η έννοια του confusion matrix, γνωστό και ως πίνακα λαθών. Ο confusion matrix είναι ένας ειδικός πίνακας, που επιτρέπει την επίβλεψη της επίδοσης ενός αλγορίθμου machine learning. Κάθε γραμμή του πίνακα αντιπροσωπεύει τις προβλέψεις και κάθε στήλη τις πραγματικές κατηγορίες των δειγμάτων (εικόνα 12).

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Εικόνα 12. Confusion matrix layout.

Στον πίνακα με **P συμβολίζονται οι θετικές κατηγορίες**, η έκκριση της πρωτεΐνης στην περίπτωση μας και με **N οι αρνητικές κατηγορίες**, η μη έκκριση της πρωτεΐνης. Απαραίτητο στάδιο για τη δημιουργία του πίνακα είναι ο ορισμός ενός threshold, η τιμή του οποίου ορίζει τη θετική ή την αρνητική πρόβλεψη. Ειδικότερα, πάνω από το **threshold** η πρόβλεψη του αλγορίθμου (πιθανότητα) χαρακτηρίζεται ως θετική και κάτω από το threshold ορίζεται ως αρνητική. Όσον αφορά στο περιεχόμενο του πίνακα, τα δείγματα που βρίσκονται στο πάνω αριστερά τεταρτημόριο ονομάζονται **True Positives** γιατί η πρόβλεψη και συνάμα η πραγματική κατηγορία είναι θετικές. Εν συνεχεία, στο κάτω δεξά τεταρτημόριο βρίσκονται τα **True Negatives** δείγματα, στα οποία η πρόβλεψη και συνάμα η πραγματική κατηγορία είναι αρνητικές. Στο κάτω αριστερά τεταρτημόριο βρίσκεται ο αριθμός των δειγμάτων, τα οποία προβλέπονται ως θετικά (αλλά στην πραγματικότητα είναι αρνητικά) (False Negatives) εν αντιθέσει με το πάνω δεξά τεταρτημόριο, τα δείγματα του οποίου στην πραγματικότητα είναι θετικά αλλά προβλέπονται ως αρνητικά (False Positives) [16,17].
Από τον **confusion matrix** μπορούν να εξαχθούν αρκετοί χαρακτηριστικοί δείκτες που περιγράφουν την **επίδοση ενός αλγορίθμου machine learning:**

1. Accuracy (ACC)

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

Τα False Positives και False negatives δείγματα είναι οι λάθος προβλέψεις του αλγορίθμου. Συνεπώς,, η ακρίβεια είναι το ποσοστό των σωστών προβλέψεων του αλγορίθμου.

2. Omission rate

$$Omission\,rate = 1 - ACC \quad (4)$$

Είναι το ποσοστό των λάθος προβλέψεων του αλγορίθμου.

3. Sensitivity or Recall

$$Sensitivity = \frac{TP}{TP + FN} \quad (5)$$

Η ευαισθησία (sensitivity) του αλγορίθμου είναι ο λόγος των σωστών θετικών προβλέψεων προς όλα τα θετικά δείγματα. Εν ολίγοις, η ευαισθησία ποσοτικοποιεί την ικανότητα του αλγορίθμου να αποφύγει τις False Negatives προβλέψεις.

4. Specificity

$$Specificity = \frac{TN}{TN + FP} \quad (6)$$

Εκφράζει την ικανότητα του αλγορίθμου να αποφύγει τις λάθος θετικές προβλέψεις (False Positives).

5. Precision

$$Precision = \frac{TP}{TP + FP} \quad (7)$$

Εκφράζει την ακρίβεια ή τη θετική του προβλεπτική ικανότητα.

6. Kappa

$$Kappa = \frac{observed\ accuracy - expected\ accuracy}{1 - expected\ accuracy} \quad (8)$$

$$observed\ accuracy = ACC \quad (9)$$

$$expected\ accuracy = \frac{\frac{(TP + FP) * (TP + FN)}{N} + \frac{(FN + TN) * (FP + TN)}{N}}{N} \quad (10)$$

Με **N το συνολικό αριθμό των δειγμάτων**. Ο δείκτης **expected accuracy** εκφράζει την ακρίβεια που θα είχε ένας οποιοδήποτε τυχαίος αλγόριθμος πάνω στον **confusion matrix** με βάση την κατανομή των κατηγοριών στα δεδομένα. Ουσιαστικά, το **Kappa statistic** είναι ένας καλύτερος δείκτης για την ακρίβεια, στην περίπτωση που έχουμε μη ισορροπημένες κατηγορίες-κλάσεις στα δεδομένα, καθώς λαμβάνει υπόψιν του την κατανομή των ίδιων των δεδομένων [18].

7. Matthews Correlation Coefficient (MCC)

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \qquad (11)$$

Ο δείκτης MCC παίρνει τιμές από -1 έως 1 και είναι ο πιο χρήσιμος δείκτης ποιότητας ενός αλγορίθμου όταν οι κλάσεις των δεδομένων δεν είναι ισορροπημένες.

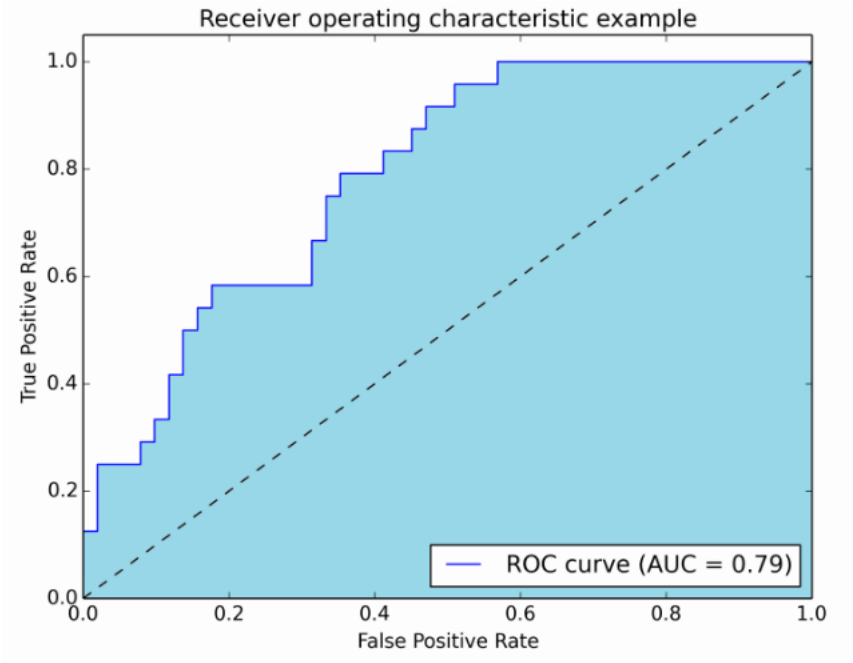
8. False Positive Rate (FPR)

$$FPR = \frac{FP}{FP + TN} \qquad (12)$$

Εκφράζει το λόγο των λάθος θετικών προβλέψεων, προς όλα τα πραγματικά αρνητικά δείγματα. Όσο μεγαλύτερος είναι ο δείκτης FPR, τόσο πιο πολλά αρνητικά δείγματα θα κατηγοριοποιηθούν λάθος.

9. Area Under the Receiver Operating Characteristic Curve (AUROC)

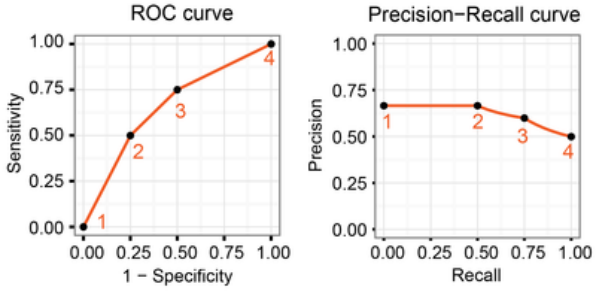
Η καμπύλη ROC παράγεται τοποθετώντας στον y άξονα τον δείκτη sensitivity και στον x άξονα τον δείκτη FPR για διαφορετικες τιμές κατωφλιού κατηγοριοποίησης (threshold). Συνεπώς, όσο μεγαλύτερη είναι η επιφάνεια κάτω από την καμπύλη, τόσο καλύτερα δουλεύει ο αλγόριθμος κατηγοριοποίησης (εικόνα 13).



Εικόνα 13. ROC curve

10. Area Under the Precision Recall Curve (AUPRC)

Η καμπύλη Precision-Recall (Sensitivity) δημιουργείται από ζευγάρια τιμών για διαφορετικά thresholds. Για το χαρακτηρισμό της καμπύλης, όσο πιο κοντά στην πάνω και δεξιά πλευρά βρίσκεται η καμπύλη, τόσο πιο καλή επίδοση έχει ο αλγόριθμος κατηγοριοποίησης (εικόνα 14) [19].



Εικόνα 14. Σύγκριση καμπύλης ROC με PRC.

4.2. Χαρακτηρισμός των μεθόδων πρόβλεψης

Με τη χρήση των προαναφερθέντων δεικτών θα κριθεί η επίδοση των δεδομένων προγραμμάτων πρόβλεψης, συγκεκριμένα των προγραμμάτων SignalP, TMHMM, Phobius, WolfPSORT. Για αυτό το λόγο, ως είσοδος στα προγράμματα δόθηκε το σετ δεδομένων, το οποίο μετασχηματίστηκε καταλλήλως και αποτελείται από 12724 πρωτεΐνες. Πρόκειται για πρωτεΐνες, για τις οποίες γνωρίζουμε εάν περιέχουν σηματοδοτικές αλληλουχίες και εάν εκκρίνονται από το κύτταρο (ground truth).
Με την έξοδο των προγραμμάτων και με τις αληθινές κατηγορίες των δεδομένων κατασκευάζεται για κάθε πρόγραμμα ο **confusion matrix** και έπειτα υπολογίζονται οι δείκτες επίδοσης του.

A. SignalP

Έξοδος του προγράμματος SignalP είναι η πιθανότητα που δίδεται ώστε η εκάστοτε πρωτεΐνη να περιέχει σηματοδοτικό πεπτίδιο. Συνεπώς, για τη δημιουργία του confusion matrix ως threshold χρησιμοποιήθηκε το 0.5, όπως ακριβώς προτείνεται και από τους δημιουργούς του προγράμματος.

Table 5: SignalP confusion matrix

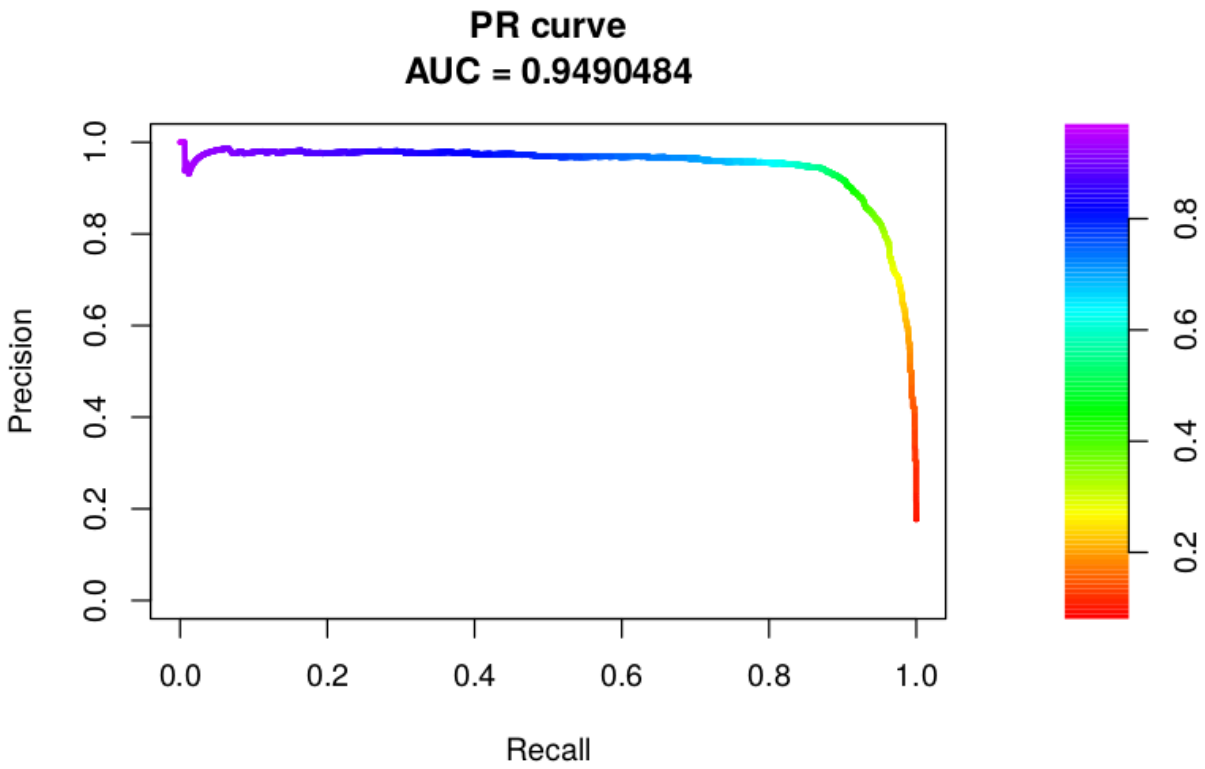
	0	1
0	10299	245
1	159	2021

Πίνακας 5. Confusion matrix of SignalP.

Table 6: SignalP Performance

threshold	AUC	omission.rate	sensitivity	specificity	prop.correct	Kappa	precsig	mccSigP
0.5	0.9383381	0.10812	0.89188	0.9847963	0.968249	0.8899043	0.9270642	0.8901488

Πίνακας 6. Επιδόσεις του προγράμματος SignalP



Εικόνα 15. Καμπύλη Precision-Recall για το πρόγραμμα SignalP

Το πρόγραμμα έχει ικανοποιητική καμπύλη (PRC) και ο δείκτης MCC έχει τιμή 0.89, γεγονότα που καθιστούν την έξοδο του αλγορίθμου πολύ αξιοπίστη για να χρησιμοποιηθεί ως χαρακτηριστική μεταβλητή για την πρόβλεψη έκκρισης ενός δείγματος, στο οποίο δεν είναι γνωστό εάν περιέχεται σηματοδοτική αλληλουχία ή όχι.

Β. TMHMM

Το πρόγραμμα TMHMM προβλέπει την ύπαρξη περιοχών μεμβράνης σε μία πρωτεΐνη με δεδομένη την αλληλουχία των αμινοξέων της. Η έξοδος του προγράμματος είναι 0 εάν η πρωτεΐνη δεν περιέχει περιοχή μεμβράνης και 1 εάν περιέχει. Αντίθετα δηλαδή με την έξοδο του SignalP, η έξοδος του TMHMM είναι λογική μεταβλητή με δεδομένο threshold = 0.5, καθορισμένο από το δημιουργό του προγράμματος. Έτσι, δυστυχώς δε δύναται η ευκαιρία να κατασκευαστούν καμπύλες λειτουργίας των προγραμμάτων ROC και PRC, έχοντας ως δοσμένες διαφορετικές τιμές στο threshold. Συγκεκριμένα, ο confusion matrix του προγράμματος για τα δεδομένα πρωτεϊνων και οι επιδόσεις του φαίνονται παρακάτω.

TMHMM confusion matrix

	0	1
0	8392	277
1	321	3734

TMHMM Performance

threshold	AUC	omission.rate	sensitivity	specificity	prop.correct	Kappa	prectm	mcctm
0.5	0.9470492	0.0690601	0.9309399	0.9631585	0.9530022	0.8914597	0.9208385	0.8914882

Πίνακας 7. Confusion matrix και επιδόσεις του προγράμματος TMHMM

Γ. Phobius

Το πρόγραμμα Phobius προβλέπει τόσο την ύπαρξη σηματοδοτικών πεπτιδίων όσο και την ύπαρξη περιοχών μεμβράνης δεδομένης της αλληλουχίας αμινοξέων μιας πρωτεΐνης. Παρόμοια με το πρόγραμμα TMHMM, το threshold του αλγορίθμου είναι καθορισμένο και ίσο με 0.5, συνεπώς ή έξοδος του προγράμματος είναι 0 ή 1 για την ύπαρξη ή όχι σηματοδοτικών πεπτιδίων και περιοχών μεμβράνης αντίστοιχα. Οι επιδόσεις του προγράμματος φαίνονται παρακάτω.

Phobius TM confusion matrix

	0	1
0	8469	212
1	244	3799

Phobius TM Performance

threshold	AUC	omission.rate	sensitivity	specificity	prop.correct	Kappa	precphtm	mccphtm
0.5	0.9595706	0.0528546	0.9471454	0.9719959	0.9641622	0.9171668	0.9396488	0.9171823

Πίνακας 8. Επιδόσεις Phobius για την πρόβλεψη περιοχών μεμβράνης.

Phobius SP confusion matrix

	0	1
0	9838	155
1	620	2111

Phobius SP Performance

threshold	AUC	omission.rate	sensitivity	specificity	prop.correct	Kappa	precphsp	mccphsp
0.5	0.9361564	0.0684025	0.9315975	0.9407152	0.9390915	0.807419	0.7729769	0.8128637

Πίνακας 9. Επιδόσεις προγράμματος Phobius για την πρόβλεψη σηματοδοτικών πεπτιδίων

Από τους πίνακες παρατηρείται πως το πρόγραμμα Phobius έχει καλύτερες επιδόσεις από το πρόγραμμα TMHMM, όσον αφορά στην πρόβλεψη περιοχών μεμβράνης. Αντίθετα για την πρόβλεψη της ύπαρξης σηματοδοτικών πεπτιδίων οι επιδόσεις του προγράμματος είναι χειρότερες από τις επιδόσεις του SignalP.
Οι επιδόσεις του ωστόσο εξακολουθούν να είναι ικανοποιητικές προκειμένου να χρησιμοποιηθούν ως χαρακτηριστικές μεταβλητές (είσοδοι) για την πρόβλεψη της έκκρισης μίας πρωτεΐνης.

Δ. WolfPSORT

Το πρόγραμμα WolfPSORT προβλέπει την κυτταρική τοποθεσία μιας πρωτεΐνης. Η έξοδος του προγράμματος για το σκοπό της έρευνας μετατράπηκε σε 0, εάν η τοποθεσία της πρωτεΐνης είναι κάποιο οργανίδιο εντός του κυττάρου και 1 εάν η τοποθεσία της είναι εκτός κυττάρου.

WolfPSORT confusion matrix

	0	1
0	10526	133
1	1108	955

WolfPSORT Performance

threshold	AUC	omission.rate	sensitivity	specificity	prop.correct	Kappa	precwolf	mccwolf
0.5	0.8912596	0.1222426	0.8777574	0.9047619	0.9024524	0.5564911	0.4629181	0.5936988

Πίνακας 10. Επιδόσεις προγράμματος WolfPSORT.

Από τον πίνακα 10 παρατηρείται πως ο λόγος mcc είναι χαμηλός παρόλο που το πρόγραμμα έχει υψηλή ακρίβεια, sensitivity και specificity. Ως εκ τούτου, παρατηρώντας τον confusion matrix, ο χαμηλός λόγος mcc εξηγείται από τον υψηλό αριθμό False Positives. Πρόκειται για τις 1108 πρωτεΐνες, που ενώ στην πραγματικότητα είναι ενδοκυτταρικές, το πρόγραμμα τις προβλέπει ως εξωκυταρρικές. Αυτό το γεγονός, σε συνδυασμό με το γενικά χαμηλό αριθμό των positive δειγμάτων μειώνει τη θετική προβλεπτική ικανότητα του αλγορίθμου και οδηγεί στη χαμηλή τιμή του mcc.

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές | Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

5. Δημιουργία αλγορίθμου πρόβλεψης

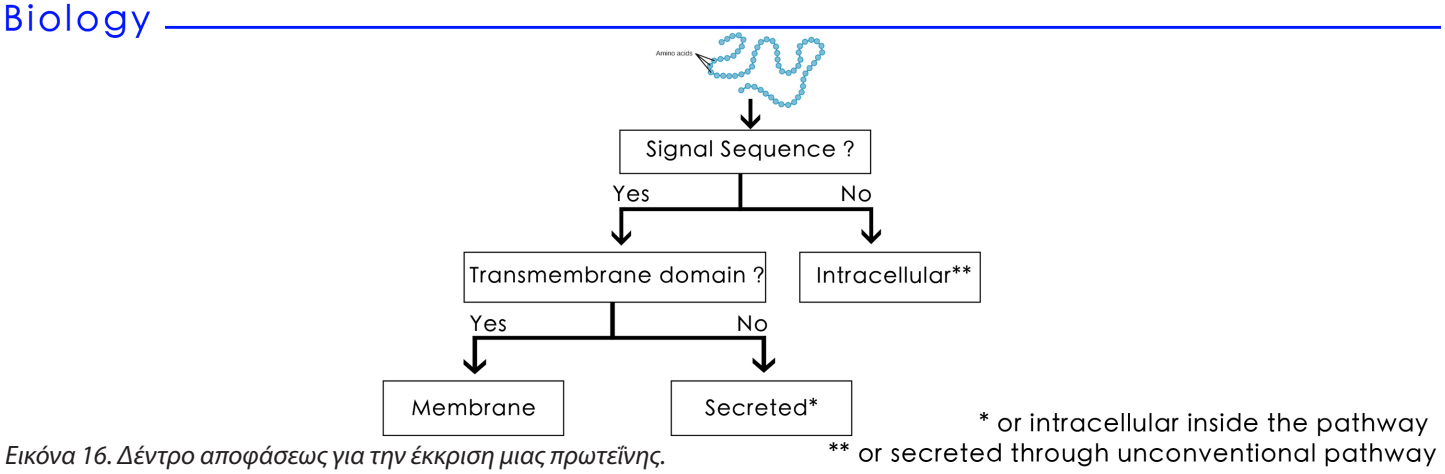
Σε αυτή την ενότητα σκοπός είναι να δημιουργηθεί ένας **αλγόριθμος ταξινόμησης**, που θα προβλέπει επιτυχώς εάν μια πρωτεΐνη εκκρίνεται ή όχι από το κύτταρο.

5.1. Υπάρχουσα μεθοδολογία

Μέχρι στιγμής, το πρόβλημα της πρόβλεψης έκκρισης μιας πρωτεΐνης λύνεται είτε:

1. Με τη χρήση ενός υπάρχοντα αλγορίθμου κατηγοριοποίησης, που χρησιμοποιεί την πληροφορία της ύπαρξης σηματοδοτικών αλληλουχιών σε μία πρωτεΐνη για να προβλέψει την τοποθεσία της:
Το πιο γνωστό και ευρέως διαδεδομένο πρόγραμμα αυτής της κατηγορίας είναι **το πρόγραμμα WolfPSORT**, που χρησιμοποιεί τη **μεθοδολογία k-NN** για να κατηγοριοποιήσει τις πρωτεΐνες στις θέσεις που εμφανίζονται. Όπως φάνηκε και στην προηγούμενη ενότητα, το πρόγραμμα εμφανίζει αρκετά δείγματα ως εξωκυτταρικά ενώ στην πραγματικότητα είναι ενδοκυτταρικά, παρουσιάζει δηλαδή αρκετά **False Positives** [20].

2. Είτε με τη δημιουργία ενός απλού δέντρου αποφάσεως, που βασίζεται στη βιολογική γνώση, χωρίς τη χρήση κάποιου αλγορίθμου τεχνητής νοημοσύνης. Με βάση αυτή τη λογική, μια πρωτεΐνη εκκρίνεται εάν περιέχει σηματοδοτικό πεπτίδιο και δεν περιέχει περιοχή μεμβράνης. Συνεπώς, το πρόβλημα απλά ανάγεται στη σωστή πρόβλεψη των σηματοδοτικών αλληλουχιών μέσα στην πρωτεΐνη. Στη βιβλιογραφία, πιο συχνή είναι η χρήση του προγράμματος SignalP για την πρόβλεψη των σηματοδοτικών πεπτιδίων σε συνδυασμό με το TMHMM, για τις περιοχές μεμβράνης [21].



Εικόνα 16. Δέντρο αποφάσεως για την έκκριση μιας πρωτεΐνης.

SignalP and TMHMM combination

	0	1
0	11399	290
1	236	799

SignalP and TMHMM combination

threshold	AUC	omission.rate	sensitivity	specificity	prop.correct	Kappa	precmodel55	mccmodel55
0.5	0.8567085	0.2662994	0.7337006	0.9797164	0.9586608	0.7298181	0.7719807	0.730099

Πίνακας 11. Επιδόσεις του απλού συνδυασμού SignalP, TMHMM για την πρόβλεψη έκκρισης.

Στόχος της παρούσας ερευνητικής εργασίας είναι να δημιουργηθεί ένας αλγόριθμος πρόβλεψης, που θα έχει καλύτερες επιδόσεις από τις υπάρχουσες μεθόδους.

5.2. Μορφή των δεδομένων για training και testing

Οι 12724 πρωτεΐνες που αποτελούν το σετ των δεδομένων δόθηκαν ως είσοδοι στα προγράμματα πρόβλεψης SignalP, TMHMM, Phobius, WolfPSORT και οι έξοδοι τους χρησιμοποιήθηκαν για τη δημιουργία του πίνακα 12.

Entry	SigP	tmhmm	phobius_sp	phobius_tm	WolfSec	Secreted
O95714	0.123	0	0	0	0	FALSE
Q9Y543	0.137	0	0	0	0	FALSE
P06865	0.916	0	1	0	0	FALSE
B2RPK0	0.128	0	0	0	0	FALSE
P26927	0.814	0	1	0	1	TRUE
Q68CP4	0.249	1	1	1	0	FALSE

Πίνακας 12. Τελική μορφή των δεδομένων (οι πρώτες 6 σειρές).

Για αυτές τις πρωτεΐνες είναι γνωστό από βιολογικά πειράματα αν εκκρίνονται ή όχι καθώς και εάν περιέχουν σηματοδοτικές αλληλουχίες. Οι στήλες του πίνακα 12 είναι:

1. Entry: Η κωδική ονομασία της κάθε πρωτεΐνης (Uniprot ID).
2. SigP: Η πιθανότητα με βάση το πρόγραμμα SignalP η πρωτεΐνη να περιέχει σηματοδοτικό πεπτίδιο.
3. tmhmm: 0 ή 1 εάν η πρωτεΐνη δεν περιέχει ή περιέχει αντίστοιχα περιοχή μεμβράνης σύμφωνα με το πρόγραμμα TMHMM.
4. phobius_sp: 0 ή 1 εάν η πρωτεΐνη δεν περιέχει ή περιέχει αντίστοιχα σηματοδοτικό πεπτίδιο σύμφωνα με το πρόγραμμα phobius.
5. phobius_tm: 0 ή 1 εάν η πρωτεΐνη δεν περιέχει ή περιέχει αντίστοιχα περιοχή μεμβράνης σύμφωνα με το πρόγραμμα phobius.
6. WolfSec: 0 ή 1 εάν η πρωτεΐνη εκκρίνεται ή όχι με βάση το πρόγραμμα WolfPSORT.
7. Secreted: 0 ή 1 εάν η πρωτεΐνη εκκρίνεται ή όχι στην πραγματικότητα (ground truth).

Ο πίνακας 12 θα αποτελεί το σύνολο των δεδομένων για την εκπαίδευση του αλγορίθμου κατηγοριοποίησης.

5.3. Μεθοδολογία αλγορίθμου κατηγοριοποίησης

Για τη **δημιουργία του αλγορίθμου κατηγοριοποίησης** θα χρησιμοποιηθεί μια βελτιωμένη έκδοση των κλασσικών δέντρων αποφάσεων για προβλήματα κατηγοριοποίησης, που ονομάζεται **τυχαία δάση (random forests)**. Απαραίτητη προϋπόθεση για την κατανόηση της μεθόδου random forest είναι η κατανόηση της μεθόδου λειτουργίας των δέντρων απόφασης/ταξινόμησης.

5.4. Δέντρα απόφασης/ταξινόμησης

Τα **δέντρα απόφασης/ταξινόμησης (Desicion/Classification trees)** αποτελούν ένα είδος ταξινομητή από τους πολλούς που έχουν επινοηθεί μέχρι στιγμής [22]. Είναι ένα ευρέως χρησιμοποιούμενο εργαλείο τόσο για την ταξινόμηση δεδομένων όσο και για την πρόβλεψη αποτελεσμάτων [23]. Σε αυτό το υποκεφάλαιο θα περιγραφεί εν συντομία, ο τρόπος με τον οποίο λειτουργεί ο συγκεκριμένος αλγόριθμος ταξινόμησης καθώς αποτελεί τη βάση για την κατανόηση της μεθόδου ταξινόμησης «Τυχαία Δάση» (random forests), που περιγράφεται στο παρακάτω υποκεφάλαιο.

Η γενική ιδέα στα δέντρα απόφασης είναι η εξής:

- Αρχικά, βασική προϋπόθεση αποτελεί το ότι κάθε **δείγμα (object/case)** του σετ δεδομένων πρέπει να μπορεί να εκφραστεί ως μια συλλογή από τις χαρακτηριστικές μεταβλητές (features) των δεδομένων.
- Η είσοδος του αλγορίθμου λαμβάνεται από κάθε δείγμα ως ένα διάνυσμα με τις τιμές των χαρακτηριστικών μεταβλητών και με την αντίστοιχη κατηγορία στην οποία ανήκει το δείγμα. Στη συγκεκριμένη περίπτωση κάθε δείγμα είναι μια πρωτεΐνη. Οι έξοδοι των προγραμμάτων **SignalP, TMHMM, Phobius, WolfPSORT** είναι οι χαρακτηριστικές μεταβλητές και η κατηγορία/κλάση του κάθε δείγματος είναι εκκρίνεται-δεν εκκρίνεται από το κύτταρο. Τα διανύσματα αυτά αποτελούν το training set και test set του ταξινομητή.
- Στη συνέχεια, ο αλγόριθμος μηχανικής μάθησης κατασκευάζει ένα σετ από κανόνες αποφάσεων με σκοπό την σωστή ταξινόμηση των δειγμάτων στις κατηγορίες τους, παρατηρώντας και συγκρίνοντας αν τα διανύσματα εισόδου, των οποίων οι τιμές είναι κοντά η μία στην άλλη, ανήκουν ή όχι στην ίδια κατηγορία..

Το σημείο όπου γίνεται η υπόθεση για μια μεταβλητή είναι αυτό, στο οποίο η μεταβλητή χωρίζεται (split) ανάμεσα σε δύο τιμές και ταυτόχρονα χωρίζει το training set σε δύο **subsets**. Η διαδικασία του split επαναλαμβάνεται σε κάθε subset, μέχρι το κάθε subset να περιλαμβάνει δείγματα που όλα ανήκουν στην ίδια κατηγορία ή μέχρι η περαιτέρω διαίρεση να μην προσφέρει παραπάνω αξία στον αλγόριθμο. Σχηματικά, στην απεικόνιση του δένδρου το σημείο που γίνεται το split ονομάζεται «node». Από το κάθε node γεννιούνται **subsets - «κλαδιά»** (branches) με την αντίστοιχη απόφαση ή ένα κλαδί το οποίο καταλήγει σε μια κλάση.Η τελική κατάληξη ενός κλαδιού σε μία κλάση είναι ένα «φύλλο» - leaf. Σχηματικά, έχει τη μορφή ενός «**αντίστροφου δέντρου**» (τα φύλλα κάτω και η ρίζα στην κορυφή), γι' αυτό και ο αλγόριθμος έχει αυτήν την ιδιαίτερη ονομασία
– «Δένδρα Αποφάσεων».
Κατά τη διαδικασία μάθησης ενός δέντρου απόφασης, καταλυτικό στοιχείο είναι το κριτήριο που καθορίζει το καλύτερο split των δεδομένων σε κάθε node [24]. Ο πιο διαδεδομένος τρόπος για να καθοριστεί το "καλύτερο" split, γνωστός και ως gini impurity [25] είναι αυτός που ποσοτικοποιεί την ομοιογένεια της μεταβλητής στόχου (κλαση-κατηγορία) στα subset που παράγονται. Ο δείκτης **gini impurity** εκφράζει τη συχνότητα με την οποία ένα δείγμα από το σετ δεδομένων θα τοποθετείτο στη λανθασμένη κατηγορία, στην περίπτωση που η ταξινόμησή του γινόταν με βάση τη στατιστική κατανομή των κατηγοριών στο subset που δημιουργεί το εκάστοτε split.
Ορίζεται ως:

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2 \qquad (12)$$

Με I = 1,2,…,J ορίζονται οι διαφορετικές κατηγορίες στο σετ των δεδομένων και με pi ορίζεται το κλάσμα δειγμάτων, που ανήκει στην κάθε κατηγορία. Συνεπώς, έχοντας ως παράδειγμα ένα σετ με 3 κλάσεις Α, Β, Γ και με 80 δείγματα και με κλάσματα Α = 19/80, Β = 21/80, Γ = 40/80 ο δείκτης gini είναι:

$$I_{Gbase} = 0.6247 \qquad (13)$$

Προκειμένου να επιλεχθεί το καλύτερο split των δεδομένων πραγματοποιείται η ακόλουθη υπόθεση: Έστω ότι στο split 1 τα δεδομένα χωρίζονται σε subset1L (Α,Β,Γ) = (16,9,0) και subset1L (Α,Β,Γ) = (3,12,40), τότε οι δείκτης gini του split είναι το σταθμισμένο άθροισμα των δεικτών gini για τα subset L, R.

$$I_{G1} = \frac{25}{80} I_{G1L} + \frac{55}{80} I_{G1R} = 0.4331 \qquad (14)$$

Σε αντίθετη περίπτωση, ένα άλλο split 2 χωρίζει τα δεδομένα σε subset2L (Α,Β,Γ) = (8,4,0) και subset2R (Α,Β,Γ) = (11,17,40), με δείκτη gini παρόμοια με τη σχέση (14) 0.5417. Ακολουθώντας την παραπάνω λογική δοκιμάζονται όλα τα πιθανά split των δεδομένων και επιλέγεται ως το καλύτερο, αυτό που δίνει το χαμηλότερο δείκτη gini impurity.

Συγκρίνοντας τα δέντρα απόφασης με άλλες μεθόδους μηχανικής μάθησης, τα δέντρα απόφασης έχουν αρκετά πλεονεκτήματα αλλά και ορισμένους σημαντικούς περιορισμούς [26].

Πλεονεκτήματα:

1. Είναι εύκολα στην κατανόηση και δίνουν τη δυνατότητα γραφικής αναπαράστασης με κατανοητό τρόπο.
2. Μπορούν να χειριστούν και αριθμητικά και λογικά (κατηγορικά) δεδομένα με την ίδια ευκολία.
3. Απαιτούν μικρή προετοιμασία των δεδομένων για την εκπαίδευση τους.
4. Αποτελούν μοντέλα που μπορούν να χαρακτηριστούν ως white boxes, δηλαδή διαθέτουμε πλήρη γνώση της εσωτερικής λειτουργίας τους.
5. Είναι εύκολο να ποσοτικοποιηθεί η επίδοσή τους.
6. Λειτουργούν ικανοποιητικά για μεγάλο όγκο δεδομένων.
7. Προσομοιάζουν τον ανθρώπινο τρόπο ταξινόμησης καλύτερα από άλλες μεθόδους.
8. Πραγματοποιούν αυτόματα επιλογή των σημαντικών μεταβλητών, με την έννοια ότι λιγότερο σημαντικά μεταβλητές δε χρησιμοποιούνται για χωρισμό των δεδομένων σε subsets.

Περιορισμοί:

1. Δεν έχουν τόσο υψηλή ακρίβεια όσο άλλες μέθοδοι,
2. Τα δέντρα απόφασης μπορεί να μην είναι στιβαρά σε αλλαγές στα δεδομένα, στα οποία εκπαιδεύτηκαν (low robustness).
3 Ο τρόπος με τον οποίο διαλέγεται το καλύτερο split δεν εγγυάται ότι δημιουργείται το ολικό βέλτιστο δέντρο (γιατί κάθε φορά διαλέγεται το καλύτερο split χωρίς να λαμβάνονται υπόψιν παλαιότερα splits [27,28]).
4. Διατρέχουν τον κίνδυνο να γίνει **overfit** στα δεδομένα που εκπαιδεύτηκαν, με αποτέλεσμα να έχουν μειωμένη προβλεπτική ικανότητα σε νέα δεδομένα [29].
5. Για δεδομένα που περιέχουν κατηγορικές μεταβλητές, το δέντρο προκύπτει ως προκατηλειμμένο (high bias) σε σχέση με τις μεταβλητές με πολλές κατηγορίες [30].

Μια βελτιωμένη και εξίσου ευρέως εφαρμοσμένη μέθοδος ταξινόμησης που απαντά στους περιορισμούς των δέντρων απόφασης είναι τα τυχαία δάση.

5.5.Τυχαία δάση (random forests)

Τα **random forests** είναι μια συλλογή από δέντρα αποφάσεων τα οποία εκπαιδεύονται σε ένα μέρος του σετ δεδομένων και λειτουργούν με τον ακόλουθο τρόπο [31]:

- Από τα δεδομένα μαθαίνονται πολλά δέντρα αποφάσεων και το κάθε ένα από αυτά δίνει μια ταξινόμηση για ένα δείγμα.
- Η τελική κλάση του δείγματος αποφασίζεται μετά από "ψηφοφορία" όλων των δέντρων.

Εύκολα προκύπτει το συμπέρασμα, ότι στα παραπάνω βήματα κύριο ρόλο παίζει το ο τρόπος με τον οποίο αναπτύσσονται τα decision trees. Κάθε δέντρο αναπτύσσεται σύμφωνα με τον ακόλουθο αλγόριθμο [32]:

1. Έστω N ο αριθμός των δειγμάτων ενός σετ δεδομένων.
2. Για κάθε δέντρο επιλέγονται n τυχαία δείγματα με εναπόθεση – **bootstrap sampling** και αποτελούν το **training set** του κάθε δέντρου.

Η μέθοδος του bootstrap sampling αποτελεί μια τεχνική συλλογής δειγμάτων από ένα **data set** και λειτουργεί ως εξής:
Έστω ότι υπάρχει ένα καλάθι με 5 μπάλες (samples) με ονόματα (labels): {A, B, C, D, E} .Από το καλάθι επιλέγεται στην τύχη μία μπάλα και καταγράφεται το όνομα της: έστω η B. Στη συνέχεια, επανατοποθετείται η μπάλα B στο καλάθι και επιλέγεται ξανά μία στην τύχη. Η διαδικασία αυτή επαναλαμβάνεται όσες φορές είναι επιθυμητό. Στο τέλος η καταγραφή των δειγμάτων που επιλέχθηκε μπορεί να μοιάζει σαν και αυτή: [B, E, D, B, C, C, A, D, E, B, A, A, E, C, E, D]. Η τεχνική αυτή ονομάζεται **bootstrap sampling ή sampling with replacement** και το τελικό δείγμα **bootstrap sample** (το οποίο είναι το input vector, που δέχεται και ξεκινάει την ανάπτυξή του κάθε ένα δένδρο).
Εφαρμόζοντας την τεχνική αυτή, σε μεθόδους ταξινόμησης, δίνεται η δυνατότητα να δημιουργηθούν από ένα **data set** περισσότερα από ένα **training sets.** Αυτός είναι και ο λόγος που χρησιμοποιείται στα **random forests,** καθώς είναι επιθυμητή η δημιουργία πολλών διαφορετικών δένδρων και ως εκ τούτου πολλών training sets (εφόσον κάθε δένδρο πρέπει να έχει το δικό του training set). Με την παραπάνω τεχνική εξασφαλίζεται ότι τα δέντρα που μαθαίνονται δεν είναι συσχετισμένα, καθώς έχουν εκπαιδευτεί σε διαφορετικά training sets. Συνεπώς, παρόλο που ένα δέντρο είναι ευαίσθητο σε θόρυβο μέσα στα δεδομένα, ο μέσος όρος πολλών δέντρων ή του "δάσους" δεν είναι.

Ένα επιπλέον χαρακτηριστικό λειτουργίας της μεθόδου έγκειται στον τρόπο που μαθαίνονται τα διαφορετικά δέντρα αποφάσεων. Στα random forests ο αλγόριθμος μάθησης των δέντρων διαλέγει σε κάθε split ένα τυχαίο δείγμα χαρακτηριστικών μεταβλητών, m σε αριθμό, για να χωρίσει το training set σε subsets. Η διαδικασία αυτή ονομάζεται feature bagging και ακολουθείται γιατί, στην περίπτωση που κάποιες μεταβλητές είναι ισχυροί ταξινομητές, τότε αυτές θα διαλεχτούν σε πολλά από τα αναπτυσσόμενα δέντρα με αποτέλεσμα αυτά να είναι συσχετισμένα [33]. Ο αριθμός m των μεταβλητών που διαλέγονται σε κάθε split αποτελεί μια απο τις πλέον σημαντικές ρυθμιστικές παραμέτρους των random forests, η οποία επηρεάζει σημαντικά τις επιδόσεις τους. Για ένα σετ δεδομένων με M μεταβλητές προτείνεται :

$$m=\sqrt{M} \qquad (15)$$

Στρογγυλοποιημένο προς τα κάτω:

Πέρα από τον αριθμό των μεταβλητών m μια επιπλέον πολύ σημαντική παράμετρος των random forests είναι η επιλογή του κριτηρίου, που καθορίζει το καλύτερο split σε κάθε node ώστε να βελτιστοποιηθεί η επίδοση του αλγορίθμου. Μια επιλογή είναι να χρησιμοποιηθεί ο δείκτης gini, που ποσοτικοποιεί την ομοιογένεια της μεταβλητής στόχου στα παραγόμενα subsets, όπως αναφέρθηκε προηγουμένως. Μια άλλη δυνατότητα είναι η χρήση της διαδικασίας extra trees κατά την οποία σε κάθε split δίνονται τυχαίες τιμές χωρισμού του training set και στο τέλος επιλέγεται το δέντρο με τις καλύτερες επιδόσεις. Αυτή η διαδικασία διαφέρει από τη χρήση του δείκτη gini στο γεγονός πως ότι δεν υπολογίζεται ένα τοπικό βέλτιστο split αλλά αντίθετα παράγεται επιπλέον ποσότητα δέντρων και επιλέγεται ο αριθμός αυτών με τις καλύτερες επιδόσεις.

5.6. Σημαντικότητα μεταβλητών

Τα random forests μπορούν να χρησιμοποιηθούν για την εξαγωγή της σημαντικότητας των χαρακτηριστικών μεταβλητών κατά την ταξινόμηση με φυσικό τρόπο. Η διαδικασία που ακολουθείται είναι η εξής [34]:

1. Έστω ένα σετ δεδομένων με N δείγματα και Xi,Yi με i=1,2,..N, χαρακτηριστικές μεταβλητές και αποκρίσεις.
2. Υπολογίζεται το out-of-bag σφάλμα του δάσους και στη συνέχεια οι τιμές της χαρακτηριστικής μεταβλητής j ανακατεύονται μεταξύ των δειγμάτων N του σετ δεδομένων. Τέλος, υπολογίζεται ξανά το σφάλμα του δάσους.
3. Η διαφορά των δύο παραπάνω σφαλμάτων αποτελεί έτσι ένα μέτρο της σημαντικότητας της j μεταβλητής, με μεταβλητές που οδηγούν σε μεγάλες διαφορές να χαρακτηρίζονται ως σημαντικές.

Με την έννοια out of bag εννούμε τα δείγματα x που δε δόθηκαν ως training set σε κάθε δέντρο, κατά τη διαδικασία του bagging, στο οποίο με εναπόθεση επιλέχθηκαν n από τα συνολικά N δείγματα.
Τα random forests δεν έχουν την ανάγκη χωρισμού του σετ δεδομένων σε training και testing για τον χαρακτηρισμό των επιδόσεων τους, επειδή κάνουν χρήση της διαδικασίας bagging. Συνεπώς,για τα random forests αρκεί ο υπολογισμός του σφάλματος στα out of bag δείγματα, στα οποία ο αλγόριθμος δεν έχει εκπαιδευτεί [35].

5.7. Πλεονεκτήματα των random forests

Παρακάτω αναφέρονται συνοπτικά τα πλεονεκτήματα των random forests, τα οποία προκύπτουν από την ανάλυση λειτουργίας της μεθόδου και απαντούν στους περιορισμούς των δέντρων αποφάσεων:

1. Το σφάλμα γενίκευσης είναι αρκετά περιορισμένο, από τη στιγμή που αναπτύσσεται ένας πολύ μεγάλος αριθμός δέντρων, με αποτέλεσμα να είναι απίθανο να παρουσιαστεί το πρόβλημα της υπέρ-εκπαίδευσης (over fitting).
2. Η τυχαία επιλογή των μεταβλητών πρόβλεψης (υπεύθυνες για το splitting στα nodes) μειώνει τη συσχέτιση των μεγάλων δένδρων, κάτι που κάνει την όλη μέθοδο αρκετά αμερόληπτη (low bias).
3. Η υλοποίηση των random forests είναι εύκολα παραλληλοποιήσιμη.

5.8. Υλοποίηση της μεθόδου random forest

Για τη δημιουργία του αλγορίθμου random forest χρησιμοποιήθηκε το πακέτο caret και η υλοποίηση έγινε στην γλώσσα προγραμματισμού R.
Ώς σετ δεδομένων για την εκπαίδευση και τον έλεγχο των επιδόσεων του δάσους θα χρησιμοποιηθεί το μετασχηματισμένο σετ δεδομένων που αναφέρθηκε στην προηγούμενη ενότητα και το οποίο παρουσιάζεται ξανά παρακάτω:

Entry	SigP	tmhmm	phobius_sp	phobius_tm	WolfSec	Secreted
O95714	0.123	0	0	0	0	FALSE
Q9Y543	0.137	0	0	0	0	FALSE
P06865	0.916	0	1	0	0	FALSE
B2RPK0	0.128	0	0	0	0	FALSE
P26927	0.814	0	1	0	1	TRUE
Q68CP4	0.249	1	1	1	0	FALSE

Πίνακας 12. Οι πρώτες 6 σειρές του σετ δεδομένων.

Έτσι, το input vector για την εκπαίδευση του αλγορίθμου είναι [SigP,tmhmm,phobius_sp,phobius_tm,WolfSec] και η μεταβλητή στόχος που πρέπει να προβλευτεί είναι η Secreted.

Παρόλο που κατά την δημιουργία ενός random forest μπορεί να χρησιμοποιηθεί το out-of-bag error, για το χαρακτηρισμό των επιδόσεων του και για τη βελτιστοποίηση των παραμέτρων του, στην προκειμένη περίπτωση επιλέχθηκε να χρησιμοποιηθεί η γνωστή τεχνική του k-fold cross-validation.

Με τη διαδικασία του k-fold cross-validation, το σύνολο των δεδομένων χωρίζεται τυχαία σε k τμήματα και ο αλγόριθμος εκπαιδεύεται k φορές, χρησιμοποιώντας σε κάθε επανάληψη ένα τμήμα των δεδομένων για την εκτίμηση των επιδόσεων του και τα υπόλοιπα για την εκπαίδευση του. Με τη διαδικασία του cross-validation μειώνεται το σφάλμα γενίκευσης του αλγορίθμου γιατί ο αλγόριθμος δεν κάνει overfit στα δεδομένα εκπαίδευσης.

5.9. Parameter tuning

Οι παράμετροι του random forest προς βελτιστοποίηση είναι [36]:

1. mtry = ο αριθμός των τυχαίων μεταβλητών που επιλέγονται σε κάθε split στα δέντρα αποφάσεων, με σύνολο τιμών mtry = [3,4,5].
2. splitrule = το κριτήριο με βάση το οποίο γίνεται το split σε κάθε node, με 2 πιθανές περιπτώσεις.
   α. Gini, όταν χρησιμοποιείται ο δείκτης gini
   β. Extratrees, όταν χρησιμοποιείται η τυχαία μέθοδος για το split.

Ο αριθμός των δέντρων δεν αποτελεί παράμετρο προς βελτιστοποίηση, γιατί έχει αποδειχτεί ότι ενώ η επίδοση του δάσους αυξάνει καθώς αυξάνεται ο αριθμός των δέντρων, από ένα σημείο και μετά,η επίδοση παραμένει σταθερή, όσο και εάν αυξάνονται τα δέντρα. Για αυτό το λόγο επιλέχτηκε το δάσος να αποτελείται από 500 δέντρα.

Το κριτήριο βελτιστοποιήσης με βάση το οποίο επιλέχτηκαν οι τιμές των παραμέτρων είναι η μεγιστοποίηση του δείκτη Kappa. Ο δείκτης Kappa επιλέχτηκε καθώς αποτελεί έναν καλό δείκτη της επίδοσης ενός αλγορίθμου ταξινόμησης σε σετ δεδομένων, όπου η μια κλάση κυριαρχεί. Στην περίπτωση μας, κυρίαρχη κλάση είναι η μη έκκριση μιας πρωτεΐνης.

Παρακάτω, στον πίνακα 13 παρουσιάζεται η τιμή του kappa για όλους του πιθανούς συνδυασμούς mtry, splitrule.

mtry	splitrule	Accuracy	Kappa	AccuracySD	KappaSD
3	gini	0.9700558	0.8119388	0.0046846	0.0278493
3	extratrees	0.9695058	0.8084363	0.0053067	0.0329001
4	gini	0.9698205	0.8112520	0.0050174	0.0304674
4	extratrees	0.9702130	0.8126153	0.0050369	0.0303121
5	gini	0.9625898	0.7628107	0.0058645	0.0357590
5	extratrees	0.9636899	0.7709488	0.0057303	0.0354337

Πίνακας 13. Επιδόσεις random forest για το εύρος τιμών των παραμέτρων.

Από τον πίνακα 13 εξάγεται το συμπέρασμα ότι η βέλτιστη τιμή του kappa επιτυγχάνεται για mtry = 4 και splitrule = extratrees. Εκτός από το kappa, για αυτές τις τιμές των παραμέτρων γίνεται βέλτιστη και η ακρίβεια του αλγορίθμου.

5.10. Επιδόσεις του αλγορίθμου

Στον πίνακα 14 παρουσιάζεται ο confusion matrix του αλγορίθμου και στον πίνακα 15 παρουσιάζονται οι δείκτες επίδοσης του αλγορίθμου.

	0	1
0	11461	132
1	174	957

Πίνακας 14. Random forest confusion matrix

threshold	AUC	omission.rate	sensitivity	specificity	prop.correct	Kappa	precmodel5	mccmodel5
0.5	0.9319165	0.1212121	0.8787879	0.9850451	0.975951	0.8489936	0.8461538	0.849176

Πίνακας 15. Επιδόσεις random forest

Οι επιδόσεις του random forest είναι πολύ υψηλές, με χαμηλό αριθμό false positives και false negatives και mcc = 0.849176.
Η τιμή του mcc και γενικά η επίδοση του random forest είναι καλύτερη από τις υπάρχουσες προαναφερθείσες μεθοδολογίες, που χρησιμοποιούνται για την πρόβλεψη της έκκρισης μια πρωτεΐνης.

5.11. Σημαντικότητα μεταβλητών

```
## ranger variable importance
##
##            Overall
## WolfSec     100.00
## phobius_tm   62.57
## SigP         59.33
## phobius_sp   23.01
## tmhmm         0.00
```

Πίνακας 16. Random forest variable importance

Με βάση τον πίνακα 16, το random forest δίνει μεγάλη σημασία στην έξοδο του προγράμματος WolfPSORT, παρόλο που η πρόβλεψη είναι βελτιωμένη με την επιπλέον χρήση των εξόδων των άλλων προγραμμάτων. Για την πληροφορία της ύπαρξης σηματοδοτικού πεπτιδίου, το random forest χρησιμοποιεί την έξοδο και των δύο προγραμμάτων SignalP και Phobius, δίνοντας μεγαλύτερο βάρος στην πρόβλεψη του SignalP. Όσον αφορά στις περιοχές μεμβράνης, το random forest χρησιμοποιεί την πληροφορία μόνο από το πρόγραμμα phobius και όχι από το πρόγραμμα tmhmm. Η μη χρήση της μεταβλητής tmhmm εκφράζει την πλήρη συσχέτιση της με την μεταβλητή phobius_tm.

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές | Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

6. Ανάλυση των στοιχείων επαλήθευσης της κυτταρικής τοποθεσίας μια πρωτεΐνης

Μέσα στο σύνολο των δεδομένων πρωτεϊνών, το οποίο λήφθηκε από τη βάση δεδομένων uniprot και συγκεκριμένα στη στήλη subcellular location υπάρχουν ορισμένες πρωτεΐνες, για τις οποίες οι έρευνες δε συμφωνούν σχετικά με το εάν εκκρίνονται ή όχι.
Για παράδειγμα η πρωτεΐνη με κωδικό A1E959 έχει τοποθεσία (εικόνα 17):

```
[1] "SUBCELLULAR LOCATION: Secreted {ECO:0000250|UniProtKB:Q3HS83}. Cytoplasm {ECO:0000269|PubMed:25911094}.
 Nucleus {ECO:0000269|PubMed:25911094}."
```

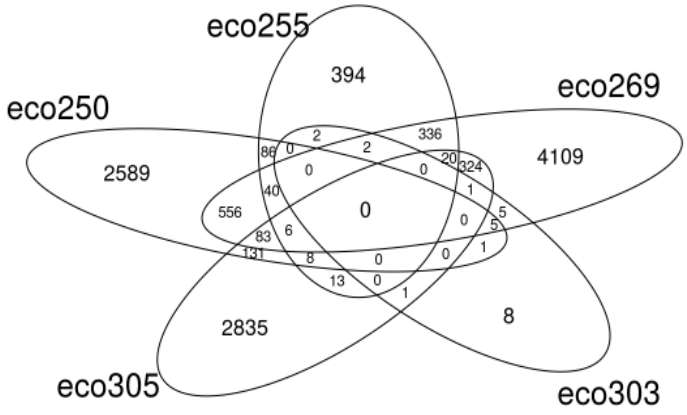
Παρατηρείται πως υπάρχουν στοιχεία, τα οποία τοποθετούν τη συγκεκριμένη πρωτεΐνη εντός του κυττάρου και στοιχεία, τα οποία υποστηρίζουν την έκκριση της από το κύτταρο. Κατά τη διαδικασία μετασχηματισμού των δεδομένων για τη δημιουργία του training set για το random forest, αποφασίστηκε η διαγραφή τέτοιων cases αφού για την εκπαίδευση του αλγορίθμου είναι απαραίτητη η κατοχή "καθαρών" κατηγοριών/κλάσεων (εκκρίνεται-δεν εκκρίνεται).
Στο παράδειγμα της πρωτεΐνης με κωδικό A1E959, με περαιτέρω διερεύνηση παρατηρείται πως κάθε πιθανή τοποθεσία συνοδεύεται από ένα κωδικό ECO (evidence code), που περιγράφει την προέλευση των στοιχείων για τον εκάστοτε χαρακτηρισμό της τοποθεσίας της πρωτεΐνης.
Τα διάφορα ECO που υπάρχουν στο σύνολο των δεδομένων είναι:

1. ECO255, με βάση την αλληλουχία αμινοξέων της πρωτεΐνης.
2. ECO250, με βάση την ομοιότητα της αλληλουχίας αμινοξέων της πρωτεΐνης με άλλες πρωτεΐνες.
3. ECO269, με βάση τα δημοσιευμένα πειραματικά αποτελεσμάτα.
4. ECO305, με βάση την επιστημονική γνώση ενός expert κριτή.
5. ECO303, με βάση τις αναφορές σε επιστημονικά άρθρα.

Στόχος σε αυτή την ενότητα, είναι η περαιτέρω διερεύνηση των πρωτεϊνών, που περιέχουν αντικρουόμενες πληροφορίες σχετικά με τη θέση τους εντός ή εκτός του κυττάρου βασιζόμενες σε διαφορετικά ECO. Η διερεύνηση αυτή θα γίνει χρησιμοποιώντας την έξοδο του αλγορίθμου πρόβλεψης ως επιπλέον στοιχείο, με σκοπό να κριθεί η δύναμη του κάθε ECO.

6.1. Μετασχηματισμός των δεδομένων

Για τους σκοπούς της διερεύνησης των ECO ακολουθήθηκε μια διαφορετική διαδικασία μετασχηματισμού των δεδομένων από την αρχική. Αρχικά, αφαιρέθηκαν από το σύνολο των δεδομένων οι πρωτεϊνες/cases που δε συμπεριλαμβάνουν τον αντίστοιχο κωδικό ECO για την κυτταρική τους τοποθεσία. Στη συνέχεια, το αρχικό σύνολο χωρίστηκε σε υποσύνολα, το περιεχόμενο και η αλληλοεπικάλυψη των οποίων παρουσιάζονται στην εικόνα 18:



Εικόνα 18. Αλληλοεπικάλυψη των διάφορων ECO.

Παρατηρείται πως τα περισσότερα δεδομένα περιλαμβάνουν τους κωδικούς ECO250, ECO255 και ECO305. Στη συνέχεια, διερευνώντας τις πρωτεΐνες που περιλαμβάνουν τοποθεσίες με περισσότερα από ένα στοιχεία ECO, προκύπτει πως 72 πρωτεΐνες περιέχουν αντικρουόμενες τοποθεσίες από διαφορετικά ECO (πίνακας 17).

Entry	Location	Evidence	Domain	hasSP	hasTM	Secreted
A1E959	cytoplasm, nucleus / secreted	eco:0000269 / eco:0000250	SP only	1	0	FALSE / TRUE
A5D8T8	secreted / endoplasmic reticulum, golgi apparatus, endosome	eco:0000269 / eco:0000305	SP only	1	0	TRUE / FALSE
O14638	secreted / membrane, single-pass type ii membrane protein	eco:0000269 / eco:0000305	Mem. only	0	1	TRUE / FALSE
O43866	secreted / cytoplasm	eco:0000269 / eco:0000250	SP only	1	0	TRUE / FALSE
O60469	isoform short: secreted / isoform long: cell membrane, single-pass type i membrane protein, cell projection, axon, cell junction, synapse	eco:0000305 / eco:0000250	SP and Mem.	1	1	TRUE / FALSE
O75356	secreted / endoplasmic reticulum	eco:0000269 / eco:0000250	SP only	1	0	TRUE / FALSE

Πίνακας 17. Οι πρώτες γραμμές του σετ δεδομένων με αντικρουόμενες τοποθεσίες.

Οι 72 αυτές πρωτεΐνες αφαιρέθηκαν από το σετ των δεδομένων ώστε να γίνει η διερεύνηση τους.

6.2. Νέα εκπαίδευση του random forest

Μετά τη δημιουργία του νέου σετ δεδομένων, επαναλαμβάνεται η εκπάιδευση του random forest, με την ίδια λογική όπως και στην προηγούμενη ενότητα. Παρακάτω, φαίνονται οι επιδόσεις του και ο confusion matrix του χρησιμοποιώντας την τεχνική του cross-validation (Πίνακας 18,19).

	0	1
0	10497	146
1	196	651

Πίνακας 18. Confusion matrix του random forest για τα νέα δεδομένα.

threshold	AUC	omission.rate	sensitivity	specificity	prop.correct	Kappa	precmodel6	mccmodel6
0.5	0.8992417	0.183187	0.816813	0.9816703	0.970235	0.7759575	0.768595	0.7763741

Πίνακας 19. Επιδόσεις του random forest με 10-fold cross-validation στο νέο σετ δεδομένων.

Παρατηρείται πως οι επιδόσεις του random forest είναι χειρότερες από την προηγούμενη εκδοχή. Αυτό συμβαίνει λόγω της εκπαίδευσης στο καινούριο σετ δεδομένων, το οποίο περιλαμβάνει λιγότερες περιπτώσεις πρωτεϊνών που εκκρίνονται από το προηγούμενο, με αποτέλεσμα το σύνολο των δεδομένων να είναι σε μεγαλύτερο βαθμό μη ισορροπημένο. Συγκεκριμένα, στο καινούριο σύνολο περιλαμβάνονται 797 secreted πρωτεΐνες ενώ στο προηγούμενο υπήρχαν 1089. Το γεγονός αυτό μειώνει τη θετική προβλεπτική ικανότητα του δάσους καθώς και τη συνολική του επίδοση.

6.3. Ανάλυση των αντικρουόμενων πληροφοριών

Από το σύνολο των δεδομένων αφαιρέθηκαν 72 δείγματα, για τα οποία οι κωδικοί ECO δε συμφωνούν μεταξύ τους ως προς την έκκριση της. Στην ενότητα αυτή, αυτά τα δείγματα θα επανεξεταστούν σε συνδυασμό με την πρόβλεψη του random forest για την έκκριση τους.

Η κατάσταση των αντικρουόμενων πληροφοριών παρουσιάζεται παρακάτω:

1. ECO250 vs ECO269 : 26 cases
2. ECO250 vs ECO305 : 16 cases
3. ECO269 vs ECO255 : 2 cases
4. ECO269 vs ECO305 : 27 cases
5. ECO255 vs ECO303 : 2 cases
6. ECO255 vs ECO305 : 1 cases

Εικόνα 19. Δείγματα με αντικρουόμενες τοποθεσίες απο διαφορετικούς κωδικούς.

Στη συνέχεια θα χρησιμοποιηθεί η πρόβλεψη του random forest για αυτές τις πρωτεΐνες και θα διερευνηθεί το ποσοστό συμφωνίας της πρόβλεψης με κάθε κωδικό ECO, σε καθεμία από τις 6 περιπτώσεις, που οι κωδικοί δε συμφωνούν μεταξύ τους για την έκκριση ή μη της πρωτεΐνης.

1. ECO250 vs ECO269

Σε 12/26 cases η πρόβλεψη συμφωνεί με τον κωδικό 269 ενώ σε 14/26 δείγματα συμφωνεί με τον κωδικό 250. Συνεπώς, δε δύναται η εξαγωγή κάποιου συμπεράσματος, καθώς τα ποσοστά είναι πολύ κοντά το ένα στο άλλο. Ιδανικά, το ποσοστό συμφωνίας με τον κωδικό 269 θα έπρεπε να είναι μεγαλύτερο, καθώς ο κωδικός 269 βασίζεται σε πειραματικά δεδομένα και θεωρείται ο πιο αξιόπιστος.

2. ECO305 vs ECO250

Σε 10/16 cases η πρόβλεψη του random forest συμφωνεί με τον κωδικό 250, ενώ σε 6/16 συμφωνεί με τον κωδικό 305. Υπενθιμίζεται, πως ο κωδικός 305 βασίζεται στην επιστημονική γνώση ενός expert κριτή, ενώ ο κωδικός 250 βασίζεται στη μεγάλη ομοιότητα της αλληλουχίας αμινοξέων της πρωτεΐνης με μια άλλη, της οποίας η τοποθεσία είναι πειραματικά εξακριβωμένη. Εκ πρώτης όψεως, κανένας από τους δύο κωδικούς δε δύναται να χαρακτηριστεί ως πιο αξιόπιστος. Η συμφωνία του random forest σε μεγάλο ποσοστό με τον κωδικό 250 εξηγείται από τη χρήση της αλληλουχίας αμινοξέων και πιο συγκεκριμένα από τις σηματοδοτικές αλληλουχίες, ως χαρακτηριστικές μεταβλητές πρόβλεψης.

3. ECO269 vs ECO255

Οι περιπτώσεις αυτές είναι μόνο δύο, άρα κανένα σημαντικό συμπέρασμα/εξήγηση δε δύναται να εξαχθεί.

4. ECO269 vs ECO305

Από τις συνολικά 27 αυτές αντικρουόμενες περιπτώσεις, η πρόβλεψη συμφωνεί με τον κωδικό 269 19/27 και με τον κωδικό 305, 8/27 φορές. Γενικά, ο κωδικός 269, που βασίζεται σε πειραματικά δεδομένα θεωρείται ο πιο αξιόπιστος.

5. ECO255 vs ECO303

Οι περιπτώσεις αυτές είναι μόνο δύο, άρα κανένα σημαντικό συμπέρασμα/εξήγηση δε δύναται να εξαχθεί.

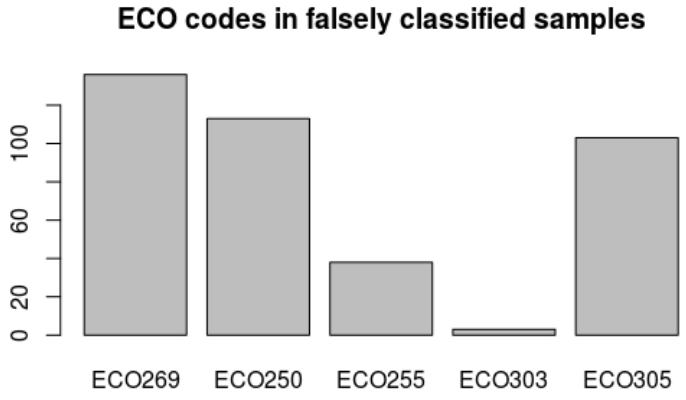
6. ECO255 vs ECO305

Η περίπτωση είναι μόνο μια, άρα κανένα σημαντικό συμπέρασμα/εξήγηση δε δύναται να εξαχθεί.

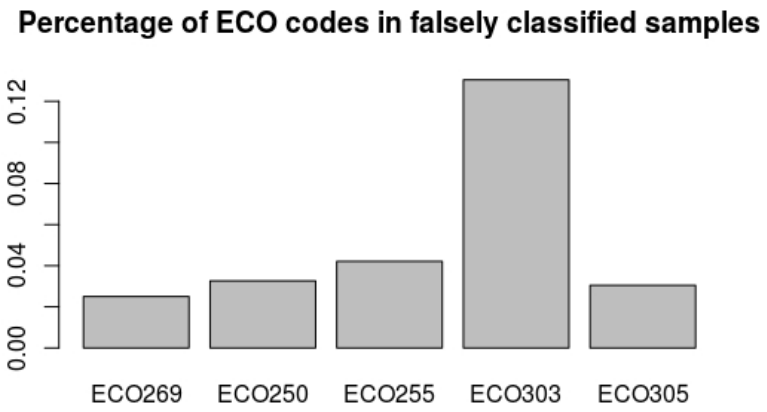
Από την παραπάνω διερεύνηση, αξιοσημείωτο είναι το συνολικό χαμηλό ποσοστό συμφωνίας του αλγορίθμου random forest, με τον κωδικό ECO305. Δεδομένων των 44 συνολικά εμφανίσεων του κωδικού, η πρόβλεψη συμφωνεί με τον κωδικό μόνο 13 φορές. Εντούτοις, το μεγαλύτερο ποσοστό συμφωνίας του αλγορίθμου εμφανίζεται με τον κωδικό 269, γεγονός θετικό, αφού ο ECO269 βασίζεται σε πειραματικά δεδομένα, όπως αναφέρθηκε και προηγούμενως.

Στη συνέχεια, ακολούθησε η εξετάση των false positive και των false negative δειγμάτων, κατά την ταξινόμηση όσον αφορά στους κωδικούς ECO που περιλαμβάνονται σε αυτά. (εικόνα 19).

6.4. Διερεύνηση της αξιοπιστίας του κωδικού ECO305



Εικόνα 20. Συχνότητα εμφάνισης κωδικών ECO στα λάθος ταξινομημένα δείγματα.



Εικόνα 21. Ποσοστό εμφάνισης κωδικών ECO στα λάθος ταξινομημένα δείγματα.

Στην εικόνα 21 παρουσιάζεται το ποσοστό εμφάνισης των κωδικών ECO έχοντας λάβει υπόψιν το συνολικό αριθμό εμφάνισης κάθε κωδικού στο σύνολο των δεδομένων. Αξίζει να σημειωθεί πως ο κωδικός 305 παρουσιάστηκε προηγουμένως να μη συμφωνεί με τις προβλέψεις, στις περιπτώσεις όπου υπάρχουν αντικρουόμενοι κωδικού. Παρατηρώντας την εικόνα, εξάγεται το συμπέρασμα ότι ο κωδικός 305 δεν εμφανίζει μεγαλύτερο ποσοστό εμφάνισης από τους άλλους κωδικούς στα λάθος ταξινομημένα δείγματα. Συνεπώς, είναι αδύνατο να θεωρηθεί ως μη αξιόπιστος.

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές | Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

7. Ανάλυση των λανθασμένων αποτελεσμάτων

Σε αυτή την ενότητα θα αναλυθεί ο confusion matrix του πρώτου random forest που εκπαιδεύτηκε στα αρχικα δεδομένα πρωτεϊνών. Συγκεκριμένα, θα γίνει διερεύνηση των false positive και false negative δειγμάτων με σκοπό να δοθούν βιολογικές εξηγήσεις ως προς τους περιορισμούς του αλγορίθμου και να βρεθούν τρόπο βελτίωσης της λειτουργίας του. Στη συνέχεια παρουσιάζεται ξανά ο confusion matrix του αλγορίθμου που παράχθηκε με 10-fold cross-validation.

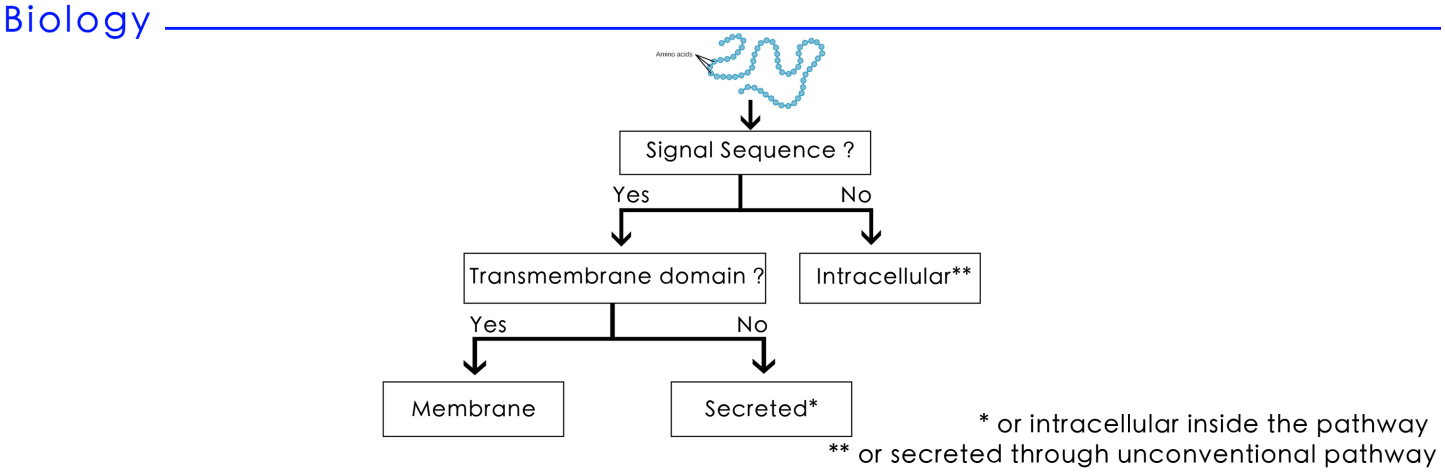
	0	1
0	11461	132
1	174	957

Πίνακας 20. Random forest confusion matrix

False positives : 174 cases τα οποία προβλέπονται από το random forest ότι εκκρίνονται ενώ στην πραγματικότητα είναι ενδοκυτταρικά.
False negatives : 130 cases τα οποία προβλέπονται από το random forest ως ενδοκυτταρικά ενώ στην πραγματικότητα εκκρίνονται.

7.1. False positives

Η λογική της διερεύνησης βασίστηκε στο αρχικό βιολογικό δέντρο για την απόφαση της έκκρισης ή μη μιας πρωτεΐνης.



Ο πρώτος αστερίσκος στην εικόνα συμβολίζει τα false positive δείγματα του δέντρου απόφασης για την έκκριση μιας πρωτεΐνης μέσω του συμβατικού μονοπατιού έκκρισης. Υπάρχουν δηλαδή στην πραγματικότητα ορισμένες πρωτεΐνες, που ενώ περιέχουν σηματοδοτικό πεπτίδιο και δεν περιέχουν περιοχή μεμβράνης, δεν εκκρίνονται από το κύτταρο αλλά επιτελούν τη λειτουργία τους εντός του εκκριτικού μηχανισμού [37]. Αυτές οι πρωτεΐνες αποτελούν ιδιαίτερες περιπτώσεις και είναι αναμενόμενο ο αλγόριθμος random forest, ο οποίος σαν χαρακτηριστικές μεταβλητές χρησιμοποιεί την ύπαρξη σηματοδοτικών αλληλουχιών, να μην μπορεί να τις ταξινομήσει σωστά.
Στόχος λοιπόν είναι να διερευνηθεί εάν τα false positive δείγματα του random forest ανήκουν σε αυτή την ειδική'κατηγορία. Για την αναγνώριση αυτών των πρωτεΐνών ακολουθήθηκαν τα παρακάτω βήματα.

1. Από τα false positive δείγματα επιλέχθηκαν αρχικά οι πρωτεΐνες οι οποίες έχουν σηματοδοτικό πεπτίδιο και δεν περιέχουν περιοχή μεμβράνης.

2. Εν συνεχεία ελέγχθηκε η κυτταρική τοποθεσία αυτών των πρωτεϊνών. Εξετάστηκε συγκεκριμένα ποιες πρωτεΐνες απο αυτές τοποθετούνται σε οργανίδια τα οποία οποία ανήκουν στο μονοπάτι έκκρισης. Τα οργανίδια αυτά είναι τα λυσοσώματα, το ενδοπλασματικό δίκτυο, το σύμπλεγμα Golgi και τα κυστίδια μεταφοράς μεταξύ αυτών των οργανιδίων.

Η παραπάνω διαδικασία εμφάνισε 133 πρωτεΐνες που περιέχουν σηματοδοτικό πεπτίδιο και όχι περιοχή μεμβράνης. Εν συνεχεία η διερεύνηση της κυτταρικής τοποθεσίας αυτών των πρωτεϊνών εμφάνισε 114 πρωτεΐνες που ανήκουν στο εκκριτικό μονοπάτι.

Entry	Location	Domain	hasSP	hasTM	Secreted
P22304	lysosome	SP only	1	0	FALSE
O14792	golgi apparatus lumen	SP only	1	0	FALSE
P35475	lysosome	SP only	1	0	FALSE
Q7Z4H8	endoplasmic reticulum lumen	SP only	1	0	FALSE
Q99538	lysosome	SP only	1	0	FALSE
P38571	lysosome	SP only	1	0	FALSE

Πίνακας 21. Οι πρώτες 6 πρωτεΐνες από τα False positive δείγματα που ανήκουν στο μονοπάτι έκκρισης.

Παρατηρείται λοιπόν ότι η αρχική υπόθεση ήταν σωστή και το random forest αδυνατεί να προβλέψει σωστά ορισμένες πρωτεΐνες που επιτελούν το έργο τους στο εκκριτικό μονοπάτι.
Οι χαρακτηριστικές μεταβλητές (προβλεπτές) για αυτές τις πρωτεΐνες παρουσιάζονται στον πίνακα 22.

Entry	SigP	tmhmm	phobius_sp	phobius_tm	WolfSec	Secreted	predca
P22304	0.864	0	1	0	1	FALSE	1
O14792	0.663	0	1	0	1	FALSE	1
P35475	0.818	0	1	0	1	FALSE	1
Q7Z4H8	0.740	0	1	0	0	FALSE	1
Q99538	0.779	0	1	0	1	FALSE	1
P38571	0.729	0	1	0	1	FALSE	1

Πίνακας 22. Χαρακτηριστικές μεταβλητές των False positive που ανήκουν στο μονοπάτι έκκρισης.

Στον πίνακα 22 με τη μεταβλητή predca συμβολίζεται η πρόβλεψη του random forest. Για αυτές τις 114 cases παρατηρούμε ότι όλες οι μεταβλητές πέρα από την πρόβλεψη του προγράμματος WolfPSORT ανταποκρίνονται στην πραγματικότητα. Η λάθος πρόβλεψη έτσι οφείλεται στην λάθος πρόβλεψη του WolfPSORT η οποία έχει 100% σημασία για την απόφαση του random forest, όπως φαίνεται και στον πίνακα 23.

```
## ranger variable importance
##
##            Overall
## WolfSec     100.00
## phobius_tm   62.57
## SigP         59.33
## phobius_sp   23.01
## tmhmm         0.00
```

Πίνακας 23. Σημαντικότητα μεταβλητών Random forest.

Αξιοσημείωτο όμως είναι ότι υπάρχουν 36 πρωτεΐνες οι οποίες ανήκουν σε οργανίδια του εκκριτικού μονοπατιού αλλά ο αλγόριθμος random forest τις προβλέπει σωστά ως ενδοκυτταρικές (Πίνακας 24, 25).

Entry	Location	Domain	hasSP	hasTM	Secreted
P06865	lysosome	SP only	1	0	FALSE
P07686	lysosome	SP only	1	0	FALSE
Q9Y4L1	endoplasmic reticulum lumen	SP only	1	0	FALSE
P48723	microsome	SP only	1	0	FALSE
P48723	endoplasmic reticulum	SP only	1	0	FALSE
Q6UW63	endoplasmic reticulum lumen	SP only	1	0	FALSE

Πίνακας 24. Πρωτεΐνες του μονοπατιού έκκρισης που προβλέπονται σωστά

Entry	SigP	tmhmm	phobius_sp	phobius_tm	WolfSec	Secreted	predca
P06865	0.916	0	1	0	0	FALSE	0
P07686	0.444	1	1	0	0	FALSE	0
Q9Y4L1	0.719	1	1	0	0	FALSE	0
P48723	0.760	1	1	0	0	FALSE	0
Q6UW63	0.846	0	1	0	0	FALSE	0
Q9H306	0.938	0	1	0	0	FALSE	0

Πίνακας 25. Χαρακτηριστικές μεταβλητές των σωστά προβλεπόμενων πρωτεϊνών.

Οι πρωτεΐνες αυτές προβλέπονται σωστά λόγω της σωστής πρόβλεψης του αλγορίθμου WolfPSORT, όπως είναι αναμενόμενο. Μία ακόμη διαφορά που θα μπορούσε να εξηγήσει τη συμπεριφορά του random forest είναι οι τιμές των χαρακτηριστικών μεταβλητών tmhmm και phobius_tm. Οι μεταβλητές αυτές εκφράζουν την πρόβλεψη ύπαρξης περιοχής μεμβράνης και παίζουν σημαντικό ρόλο στην τελική πρόβλεψη του random forest (Πίνακας 23). Από τις 36 πρωτεΐνες που προβλέπονται σωστά από τον αλγόριθμο 18 προβλέπεται ότι περιέχουν περιοχή μεμβράνης από τα προγράμματα TMHMM.

Από την παραπάνω διερεύνηση εξάγεται το συμπέρασμα ότι ο αλγόριθμος αδυνατεί να προβλέψει σωστά τις πρωτεΐνες που περιέχουν μόνο σηματοδοτικό πεπτίδιο και ανήκουν στο μονοπάτι έκκρισης. Η λάθος αυτή πρόβλεψη βασίζεται στην έλλειψη κάποιας χαρακτηριστικής μεταβλητής η οποία θα περιείχε την πληροφορία ότι η πρωτεΐνη επιτελεί το έργο της σε αυτό το μονοπάτι. Δυστυχώς μια τέτοια χαρακτηριστική μεταβλητή δεν υπάρχει. Μία δυνατότητα θα ήταν να δημιουργηθεί ένας αλγόριθμος τεχνητής νοημοσύνης που θα αναγνωρίζει αν υπάρχει κάποια σηματοδοτική αλληλουχία μέσα στην αλληλουχία αμινοξέων της πρωτεΐνης που να καθορίζει ότι η εκάστοτε πρωτεΐνη ανήκει στο μονοπάτι έκκρισης. Δυστυχώς και πάλι δεν υπάρχουν αρκετά δεδομένα τέτοιων ιδιαίτερων πρωτεΐνων ώστε να μπορεί να εκπαιδευτεί ένας τέτοιος αλγόριθμος.

Η παραπάνω διερεύνηση εμφάνισε επίσης 19 πρωτεΐνες, οι οποίες ενώ στην πραγματικότητα δεν περιέχουν σηματοδοτικό πεπτίδιο, το πρόγραμμα SignalP προβλέπει ότι έχουν. Με βάση την θετική πρόβλεψη του προγράμματος SignalP το random forest προβλέπει λανθασμένα ότι αυτές οι πρωτεΐνες εκκρίνονται από το κύτταρο, ενώ στην πραγματικότητα είναι ενδοκυτταρικές.

Περαιτέρω ανάλυση της τοποθεσίας αυτών των πρωτεϊνών αποκαλύπτει ότι οι πρωτεΐνες αυτές βρίσκονται στα μιτοχόνδρια του κυττάρου. Αυτό το γεγονός σε συνδυασμό με την λάθος πρόβλεψη του SignalP, οδηγεί στο συμπέρασμα ότι οι πρωτεΐνες αυτές περιέχουν ένα άλλο είδος πεπτιδίου, το οποίο μοιάζει με το σηματοδοτικό πεπτίδιο, αλλά δίνει στο κύτταρο την εντολή να μεταφέρει την πρωτεΐνη στα μιτοχόνδρια. Στη βιβλιογραφία τα πεπτίδια αυτά αναφέρονται ως πεπτίδια μεταφοράς σε μιτοχόνδρια [38]. Μία δυνατότητα με σκόπό τη διόρθωση της πρόβλεψης θα ήταν η χρήση ενός αλγορίθμου πρόβλεψης mtPs (mitochondria targeting peptides) ως χαρακτηριστική μεταβλητή για το random forest [39]. Η επιλογή αυτή όμως εν τέλει κρίθηκε μη αναγκαία, λόγω του μεγάλου λόγου σφάλματος που έχουν τα συγκεκριμένα προγράμματα όταν πρόκειται για τον διαχωρισμό σηματοδοτικών πεπτιδίων και πεπτιδίων μεταφοράς σε μιτοχόνδρια [39].

7.2. False negatives

Για την ανάλυση των false negative δειγμάτων ακολουθήθηκε παρόμοια διαδικασία με προηγουμένως, η οποία βασίζεται στο βιολογικό δέντρο απόφασης. Σκοπός την ανάλυσης είναι να ερευνηθεί κατά πόσο οι false negative πρωτεΐνες αποτελούν πρωτεΐνες που εκκρίνονται από το κύτταρο μέσω αντισυμβατικών μηχανισμών [40]. Από τα 132 false negative δείγματα μόνο 22 εκκρίνονται χωρίς να περιέχουν σηματοδοτικό πεπτίδιο (πίνακας 26).

Entry	Location	Domain	hasSP	hasTM	Secreted
Q9UBH0	secreted	No domain	0	0	TRUE
Q9UHA7	secreted	No domain	0	0	TRUE
P01871	isoform 1: secreted	No domain	0	0	TRUE
Q9NZH8	secreted	No domain	0	0	TRUE
Q9NZH7	secreted	No domain	0	0	TRUE
P01583	secreted	No domain	0	0	TRUE

Πίνακας 26. Οι 6 πρώτες σειρές του πίνακα με τα false negative δείγματα χωρίς σηματοδοτικό πεπτίδιο.

Entry	SigP	tmhmm	phobius_sp	phobius_tm	WolfSec	Secreted
Q9UBH0	0.168	0	0	0	1	TRUE
Q9UHA7	0.109	0	0	0	0	TRUE
P01871	0.135	0	0	0	0	TRUE
Q9NZH8	0.109	0	0	0	0	TRUE
Q9NZH7	0.138	0	0	0	0	TRUE
P01583	0.098	0	0	0	0	TRUE

Πίνακας 27. Οι πρώτες 6 σειρές του πίνακα μεταβλητών των false negative δειγμάτων χωρίς σηματοδοτικό πεπτίδιο.

Παρατηρώντας τον πίνακα 27 είναι αναμενόμενο αυτές οι πρωτεΐνες να προβλέπονται από το random forest ως ενδοκυτταρικές, αφού με βάση το SignalP δεν περιέχουν σηματοδοτικό πεπτίδιο και με βάση το WolfPSORT μόνο 3 από αυτές εκκρίνονται.
Σε αυτό το σημείο γίνεται η υπόθεση ότι αυτές οι 27 πρωτεΐνες εκκρίνονται από το κύτταρο με τη χρήση μη συμβατικών μηχανισμών έκκρισης. Για την επαλήθευση της υπόθεσης χρησιμοποιήθηκε το πρόγραμμα SecretomeP. Το πρόγραμμα αυτό δέχεται σαν είσοδο την αλληλουχία αμινοξέων της πρωτεΐνης και προβλέπει εάν η πρωτεΐνη εκκρίνεται από το κύτταρο μέσω μηχανισμών μη συμβατικής έκκρισης [41]. Στον πίνακα 28 φαίνονται τα αποτελέσματα:

# Name	NN-score	Odds	Weighted by prior	Warning
O43320	0.746	2.651	0.005	-
O75888	0.678	3.018	0.006	-
O94964	0.182	0.363	0.001	-
P01583	0.551	1.179	0.002	-
P01871	0.569	1.250	0.003	-
P27487	0.719	2.301	0.005	signal peptide predicted by SignalP
P31371	0.852	4.463	0.009	-
Q14116	0.634	1.699	0.003	-
Q16619	0.891	5.423	0.011	-
Q7L8A9	0.654	2.039	0.004	-
Q86V25	0.435	0.856	0.002	-
Q86YJ6	0.576	1.291	0.003	-
Q8N300	0.707	2.212	0.004	-
Q8WWZ1	0.565	1.303	0.003	-
Q9H4A4	0.554	1.236	0.002	-
Q9NNX1	0.472	0.943	0.002	-
Q9NP95	0.692	2.290	0.005	-
Q9NZH7	0.710	2.432	0.005	-
Q9NZH8	0.505	1.058	0.002	-
Q9UBH0	0.665	1.871	0.004	-
Q9UHA7	0.325	0.610	0.001	-

Πίνακας 28. Αποτελέσματα προγράμματος SecretomeP.

Σύμφωνα με τον πίνακα 28, 4 από τα 27 δείγματα έχουν πιθανότητα μη συμβατικής έκκρισης μεγαλύτερη από το προτεινόμενο threshold (0.5), επιβεβαιώνοντας έτσι την αρχική υπόθεση.
Σε αυτό το σημείο λοιπόν, εγείρεται ο προβληματισμός όσον αφορά την δυνατότητα χρήσης της εξόδου του προγράμματος SecretomeP ως επιπλέον χαρακτηριστική μεταβλητή στο random forest, με σκοπό να δωθεί στο δάσος η δυνατότητα να προβλέπει σωστά και τις πρωτεΐνες που εκκρίνονται μέσω μη συμβατικών μηχανισμών.
Για να εξεταστεί αυτή η δυνατότητα, δίνονται στο πρόγραμμα 50 τυχαίες πρωτεΐνες ως είσοδος οι οποίες είναι γνωστό ότι είναι ενδοκυτταρικές, και στον πίνακα 29 παρουσιάζονται τα αποτελέσματα.

Στον πίνακα 29 φαίνεται ότι 22 από τις 50 πρωτεΐνες προβλέπεται ότι εκκρίνονται, ενώ στην πραγματικότητα είναι ενδοκυτταρικές. Λόγω λοιπόν της χαμηλής αυτής ακρίβειας, το πρόγραμμα δεν επιλέχτηκε να χρησιμοποιηθεί για να ενισχύσει το random forest.

# Name	NN-score	Odds	Weighted by prior
A6NFD8	0.214	0.415	0.001
B2RPK0	0.048	0.119	0.000
O00479	0.742	2.506	0.005
O60243	0.530	1.149	0.002
O60741	0.529	1.195	0.002
P06865	0.701	2.127	0.004
P08397	0.499	1.010	0.002
P22557	0.629	1.560	0.003
P49019	0.166	0.339	0.001
P50135	0.515	1.062	0.002
P54198	0.258	0.489	0.001
P60008	0.601	1.469	0.003
P62805	0.408	0.997	0.002
Q05925	0.682	2.167	0.004
Q14774	0.310	0.615	0.001
Q16665	0.280	0.527	0.001
Q30201	0.567	1.292	0.003
Q5JVS0	0.375	0.712	0.001
Q5T447	0.352	0.676	0.001
Q5T8I9	0.507	1.020	0.002
Q5TA89	0.057	0.136	0.000
Q5VWC8	0.267	0.612	0.001
Q68CP4	0.587	1.570	0.003
Q6NXT2	0.716	2.563	0.005
Q86Z02	0.408	0.765	0.002
Q8IWW8	0.681	1.985	0.004
Q8IZP7	0.503	1.052	0.002
Q8NE63	0.424	0.794	0.002
Q8NG08	0.274	0.514	0.001
Q96DB2	0.530	1.113	0.002
Q96JB3	0.347	0.705	0.001
Q99714	0.665	1.877	0.004
Q99871	0.485	0.967	0.002
Q9BW72	0.688	2.474	0.005
Q9BXC0	0.299	0.572	0.001
Q9BXL5	0.322	0.598	0.001
Q9H2X6	0.404	0.748	0.001
Q9NRZ9	0.514	1.033	0.002
Q9NYQ3	0.531	1.141	0.002
Q9P0W2	0.359	0.693	0.001
Q9UBP5	0.423	0.906	0.002
Q9UGU5	0.535	1.179	0.002
Q9UL51	0.130	0.277	0.001
Q9UQL6	0.129	0.272	0.001
Q9Y241	0.105	0.223	0.000
Q9Y2N7	0.225	0.435	0.001
Q9Y543	0.100	0.217	0.000
Q9Y5Z7	0.486	0.983	0.002

Πίνακας 29. Αποτελέσματα Secretome P

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές | Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

8. Επίλογος-Συμπεράσματα-Περιορισμοί

Στόχος της παρούσας διπλωματικής έρευνας είναι η υπολογιστική πρόβλεψη της έκκρισης ή όχι μιας πρωτεΐνης από το κύτταρο, δεδομένης της αλληλουχίας αμινοξέων της. Η πληροφορία της έκκρισης μιας πρωτεΐνης είναι ύψιστης σημασίας τόσο για την ανακάλυψη νέων φαρμάκων όσο και για την ανακάλυψη νέων βιοδεικτών ασθένειας [42,43]. Όσον αφορά στην ανακάλυψη νέων φαρμάκων από τις 629 πρωτεΐνες, οι οποίες αποτελούν τον στόχο του συνόλου των μέχρι σήμερα εγκεκριμένων φαρμάκων, 200 εκκρίνονται από το κύτταρο [42]. Επιπλέον, μία πρωτεΐνη μπορεί να αποτελέσει χρήσιμο βιοδείκτη μιας ασθένειας μόνο όταν εκκρίνεται από το κύτταρο ώστε να μπορεί να μετρηθεί με ευκολία στα βιολογικά υγρά (αίμα, ούρα, σάλιο) [43].

Δυστυχώς όμως για ένα μεγάλο ποσοστό πρωτεΐνων δεν είναι γνωστό αν εκκρίνονται από το κύτταρο ή οχι (~20%). Από τη μοριακή βιολογία όμως είναι γνωστό ότι μια πρωτεΐνη μπορεί να εκκριθεί με δύο τρόπους. Είτε μέσω του συμβατικού μονοπατιού έκκρισης είτε μέσω μη συμβατικών μηχανισμών. Για την έκκριση μέσω του συμβατικού μονοπατιού καθοριστικό ρόλο παίζει η αλληλουχία των αμινοξέων της πρωτεΐνης και πιο συγκεκριμένα ή ύπαρξη χαρακτηριστικών περιοχών απο αμινοξέα που ονομάζονται σηματοδοτικές αλληλουχίες [44]. Έτσι, αν μια πρωτεΐνη περιέχει σηματοδοτική αλληλουχία τότε εισέρχεται στο εκκριτικό μονοπάτι και μπορεί να καταλήξει είτε εκτός του κυττάρου, είτε εντός κάποιας μεμβράνης είτε σε κάποιο οργανίδιο του μονοπατιού εντός του κυττάρου. Είναι φανερό λοιπόν ότι για την πρόβλεψη της συμβατικής έκκρισης μια πρωτεΐνης η γνώση των σηματοδοτικών αλληλουχιών που περιέχει είναι πολύ σημαντική. Για αυτό το λόγο η πρόβλεψη και αναγνώριση αυτών των ακολουθιών μέσα στην αλληλουχία αμινοξέων μιας πρωτεΐνης έχει απασχολήσει πολλά χρόνια την επιστημονική κοινότητα.

Αυτή τη στιγμή υπάρχει μια πληθώρα υπολογιστικών εργαλείων για την πρόβλεψη των σηματοδοτικών αλληλουχιών σε μία πρωτεΐνη. Τα κοινά στοιχεία όλων αυτών των προγραμμάτων είναι:
1. Δέχονται σαν είσοδο την αλληλουχία αμινοξέων μιας πρωτεΐνης.
2. Βασίζονται στη λογική της μηχανικής μάθησης. Έχουν εκπαιδευτεί δηλαδή σε δεδομένα (πρωτεΐνες) για τα οποία γνωρίζουμε απο πειράματα αλληλουχίας, εάν και τι είδος σηματοδοτικών αλληλουχιών περιέχουν (training set), με σκοπό να κάνουν προβλέψεις για πρωτεΐνες που τέτοια πληροφορία δεν υπάρχει.
Τα πιο διαδομένα προγράμματα, τα οποία αποδεδειγμένα έχουν τις καλύτερες επιδόσεις είναι [45]:
1. SignalP, Phobius για την πρόβλεψη ύπαρξης σηματοδοτικών πεπτιδίων.
2. TMHMM, Phobius για την πρόβλεψη διαμεμβρανικών περιοχών.
3. WolfPSORT για την πρόβλεψη της τοποθεσίας μια πρωτεΐνης με βάση την ομοιότητα της με άλλες πρωτεΐνες γνωστής τοποθέτησης.

Στα πλαίσια αυτής της διπλωματικής έρευνας το πρόβλημα της πρόβλεψης της έκκρισης μιας πρωτεΐνης αντιμετωπίστηκε χρησιμοποιώντας την πρόβλεψη των προαναφερθέντων προγραμμάτων για την ύπαρξη σηματοδοτικών αλληλουχιών ως ενδιάμεση πληροφορία για την τελική πρόβλεψη της έκκρισης. Αφού εξακριβώθηκαν οι καλές επιδόσεις των προγραμμάτων των οποίων οι προβλέψεις χρησιμοποιήθηκαν ως δεδομένα, δημιουργήθηκε ένας αλγόριθμος μηχανικής μάθησης ο οποίος εκπαιδεύτηκε σε αυτά με σκοπό την πρόβλεψη της έκκρισης ή όχι της πρωτεΐνης. Η υλοποίηση του αλγορίθμου έγινε με τη μέθοδο random forest σε γλώσσα προγραμματισμού R και με τη χρήση του πακέτου caret. Για τη δημιουργία του μοντέλου πρόβλεψης χρησιμοποιήθηκε 10-fold cross-validation και έγινε ρύθμιση των παραμέτρων mtry (αριθμός μεταβλητών σε κάθε split) και split rule (κριτήριου καθορισμού καλύτερου split). Το τελικό μοντέλο αποτελείται απο 500 δέντρα αποφάσεων και οι τιμές των παραμέτρων που έδωσαν το καλύτερο Kappa statistic είναι mtry = 4 και splitrule = “extra trees”.

Το random forest που εκπαιδεύτηκε παρουσιάζει πολύ καλύτερες επιδόσεις από τις υπάρχουσες μεθοδολογίες πρόβλεψης. Για την σύγκριση των μεθοδολογιών χρησιμοποιήθηκε ο δείκτης Matthews Correlation Coefficient (mcc) που εκφράζει καλύτερα την ακρίβεια ενός αλγορίθμου σε δεδομένα όπου υπερισχύει μία κλάση. Το random forest που δημιουργήθηκε έχει τιμή mcc 0.849 ενώ το πρόγραμμα WolfPSORT και ο απλός συνδυασμός του SignalP με το πρόγραμμα TMHMM έχουν mcc 0.59 και 0.74 αντίστοιχα.

Στη συνέχεια πραγματοποιήθηκε διερεύνηση ενός ιδιαίτερου υποσυνόλου των αρχικών δεδομένων, στο οποία εμφανίζονταν αντικρουόμενα στοιχεία (ground truth) σχετικά με την έκκριση των πρωτεϊνών του. Παρατηρήθηκε ότι τα αντικρουομένα αυτά στοιχεία προέρχονταν από διαφορετικά είδη ερευνών, που συμβολίζονται με τους αντίστοιχους κωδικούς ECO. Για τη διερεύνηση των κωδικών από το αρχικό σετ των δεδομένων αφαιρέθηκε το προς μελέτη υποσύνολο και όποια τυχόν δεδομένα δε συνοδεύουν την πληροφορία τους με κάποιο κωδικό ECO. Στη συνέχεια το random forest εκπαιδεύτηκε εκ νέου και οι προβλέψεις του για αυτό το υποσύνολο χρησιμοποιήθηκαν για την κρίση της ισχύος του κάθε κωδικού. Παρατηρήθηκε έτσι, ότι το χαμηλότερο ποσοστό συμφωνίας της πρόβλεψης του random forest για αυτό το υποσύνολο εμφανίζεται με τον κωδικό ECO305 13/44 cases ενώ το μεγαλύτερο ποσοστό συμφωνίας εμφανίζεται με τον κωδικό ECO269 19/27 cases, γεγονός θετικό αφού ο κωδικός 269 βασίζεται σε πειραματικά δεδομένα. Κατά την περαιτέρω διερεύνηση του κωδικού 305, δεν παρουσιάστηκε μεγαλύτερο ποσοστό εμφάνισης του στα λάθος ταξινομημένα δείγματα σε σχέση με τους υπόλοιπους κωδικούς, με αποτέλεσμα να μη δύναται να κριθεί ως μη αξιόπιστος.

Κατά τη διερεύνηση των λανθασμένων ταξινομημένων δειγμάτων από το random forest παρατηρήθηκαν ορισμένοι περιορισμοί της μεθοδολογίας, περιορισμοί οι οποίοι κυρίως σχετίζονται με τις χρησιμοποιούμενες χαρακτηριστικές μεταβλητές. Από την ανάλυση των πρωτεϊνών οι οποίες προβλέπεται ότι εκκρίνονται ενώ στην πραγματικότητα είναι ενδοκυτταρικές, παρατηρήθηκε η αδυναμία του αλγορίθμου να προβλέψει σωστά τη μη έκκριση των πρωτεϊνών. οι οποίες ανήκουν στο μονοπάτι έκκρισης. Αυτές οι πρωτεΐνες περιέχουν σηματοδοτικό πεπτίδιο στην αλληλουχία αμινοξέων τους, οπότε εισέρχονται φυσιολογικά στο μονοπάτι έκκρισης αλλά δεν εκκρίνονται ποτέ από το κύτταρο. Δυστυχώς απ'όσο γνωρίζουμε δεν υπάρχει κάποια σηματοδοτική αλληλουχία η οποία να είναι χαρακτηριστικό στοιχείο αυτών των πρωτεΐνων ώστε η ύπαρξη της να μπορεί να χρησιμοποιηθεί ως χαρακτηριστική μεταβλητή κατά την εκπαίδευση του random forest. Επιπλέον, κατά την ανάλυση εμφανίστηκαν ορισμένα δείγματα με κυτταρική τοποθεσία τα μιτοχόνδρια, τα οποία το random forest προβλέπει ότι εκκρίνονται από το κύτταρο. Πιθανότατα, το λάθος αυτό οφείλεται στο ότι το πρόγραμμα SignalP και γενικότερα τα προγράμματα πρόβλεψης σηματοδοτικών πεπτιδίων αδυνατούν να διαχωρίσουν τα σηματοδοτικά πεπτίδια με τα πεπτίδια μεταφοράς σε μιτοχόνδριο (mtp). Το ίδιο πρόβλημα όμως παρουσιάζουν και τα προγράμματα πρόβλεψης mtps και επειδή ο αριθμός πρωτεϊνών που περιέχουν mtps είναι κατά πολυ μικρότερος από τον αριθμό αυτών που περιέχουν σηματοδοτικό πεπτίδιο, επιλέχθηκε να μη συμπεριληφθούν ως χαρακτηριστικές μεταβλητές στο random forest. Από την ανάλυση των πρωτεϊνών οι οποίες στη πραγματικότητα εκκρίνονται ενώ το random forest προβλέπει ότι είναι ενδοκυτταρικές παρουσιάστηκαν ορισμένες πρωτεΐνες οι οποίες εκκρίνονται χωρίς να περιλαμβάνουν σηματοδοτική αλληλουχία. Στη συνέχεια οι πρωτεΐνες αυτές εξετάστηκε αν ανήκουν στο σύνολο των πρωτεϊνών που εκκρίνονται μέσω μη συμβατικών μηχανισμών χρησιμοποιώντας το πρόγραμμα SecretomeP. Τα αποτελέσματα του προγράμματος επιβεβαιώνουν αυτήν την υπόθεση και ταυτόχρονα επιβεβαιώνουν την αδυναμία του αλγορίθμου να προβλέψει τη μη συμβατική έκκριση πρωτεϊνών.

Είναι φανερό ότι η επίδοση του προγράμματος συνδυασμένης πρόβλεψης με τη μέθοδο random forest εξαρτάται από την επίδοση των επιμέρους προγραμμάτων πρόβλεψης που χρησιμοποιούνται ως χαρακτηριστικές μεταβλητές. Έτσι, όσο καλύτερες επιδόσεις έχουν τα προγράμματα πρόβλεψης σηματοδοτικών αλληλουχιών και όσο περισσότερη πληροφορία εξάγουν από την αλληλουχία των αμινοξέων, τόσο καλύτερη και πιο ακριβής είναι και η τελική πρόβλεψη έκκρισης ή μη της πρωτεΐνης. Ένα σημαντικό σημείο βελτίωσης θα ήταν η χρήση ενός προγράμματος για την πρόβλεψη των πρωτεΐνών που παραμένουν στο εκκριτικό μονοπάτι, και στη συνέχεια η χρήση της πρόβλεψης αυτής ως χαρακτηριστική μεταβλητή στο random forest. Τέλος αξίζει να σημειωθεί ότι ακόμη και τα πειράματα που καθορίζουν εάν μια πρωτεΐνη εκκρίνεται ή οχι έχουν μια αβεβαιότητα της τάξης του 10% η οποία πηγάζει από την πειραματική διαδικασία [46].

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές |Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

9. Πηγές | Βιβλιογραφία

1.Ponomarenko, Elena A., et al. "The size of the human proteome: the width and depth." International journal of analytical chemistry 2016 (2016).

2. Uhlén, Mathias, et al. "Tissue-based map of the human proteome." Science 347.6220 (2015): 1260419.

3. Von Heijne, Gunnar. "Signal sequences: the limits of variation." Journal of molecular biology 184.1 (1985): 99-105.

4. R.R.Wayne Wayne Albers, Chapter 2 - Cell Membrane Structures and Functions, In Basic Neurochemistry (Eighth Edition) 2012, Pages 26-39.

5. Nyathi, Yvonne, Barrie M. Wilkinson, and Martin R. Pool. "Co-translational targeting and translocation of proteins to the endoplasmic reticulum." Biochimica et Biophysica Acta (BBA)-Molecular Cell Research 1833.11 (2013): 2392-2402.

6. Luo, Guangzuo, Jian Zhang, and Wei Guo. "The role of Sec3p in secretory vesicle targeting and exocyst complex assembly." Molecular biology of the cell 25.23 (2014): 3813-3822.

7. Nickel, Walter. "Pathways of unconventional protein secretion." Current opinion in biotechnology 21.5 (2010): 621-626.

8. Meissner, Barbara, et al. "Determining the sub-cellular localization of proteins within Caenorhabditis elegans body wall muscle." PLoS One 6.5 (2011): e19937.

9. Min, Xiang Jia. "Evaluation of computational methods for secreted protein prediction in different eukaryotes." Journal of Proteomics and Bioinformatics (2010).

10. Schalkoff, Robert J. Artificial neural networks. Vol. 1. New York: McGraw-Hill, 1997.

11. Nielsen, Henrik, et al. "Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites." Protein engineering 10.1 (1997): 1-6.

12. Krogh, Anders, et al. "Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes." Journal of molecular biology 305.3 (2001): 567-580.

13. Käll, Lukas, Anders Krogh, and Erik LL Sonnhammer. "A combined transmembrane topology and signal peptide prediction method." Journal of molecular biology 338.5 (2004): 1027-1036.

14. Emanuelsson, Olof, et al. "Predicting subcellular localization of proteins based on their N-terminal amino acid sequence." Journal of molecular biology 300.4 (2000): 1005-1016.

15. Psort, I. "PSORT: a program for detecting sorting signals in proteins and predicting their subcellular localization." J. Mol. Biol 266 (1997): 594-600.

16. Bradley, Andrew P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms." Pattern recognition 30.7 (1997): 1145-1159.

17. Davis, Jesse, and Mark Goadrich. "The relationship between Precision-Recall and ROC curves." Proceedings of the 23rd international conference on Machine learning. ACM, 2006.

18. Carletta, Jean. "Assessing agreement on classification tasks: the kappa statistic." Computational linguistics 22.2 (1996): 249-254.

19. Saito, Takaya, and Marc Rehmsmeier. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets." PloS one 10.3 (2015): e0118432.

20. Cui, Juan, et al. "Computational prediction of human proteins that can be secreted into the bloodstream." Bioinformatics 24.20 (2008): 2370-2375.

21. Min, Xiang Jia. "Evaluation of computational methods for secreted protein prediction in different eukaryotes." Journal of Proteomics and Bioinformatics (2010).

22. Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." (2007): 3-24.

23. Lemon, Stephenie C., et al. "Classification and regression tree analysis in public health: methodological review and comparison with logistic regression." Annals of behavioral medicine 26.3 (2003): 172-181.

24. Rokach, Lior, and Oded Maimon. "Top-down induction of decision trees classifiers-a survey." IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 35.4 (2005): 476-487.

25. Ben-Gal, Irad, et al. "Efficient construction of decision trees by the dual information distance method." Quality Technology & Quantitative Management 11.1 (2014): 133-147.

26. James, Gareth, et al. An introduction to statistical learning. Vol. 112. New York: springer, 2013.
27. Laurent, Hyafil, and Ronald L. Rivest. "CONSTRUCTING OPTIMAL BINARY DECISION TREES IS NP-COMPLETE"=." Information Processing Letters 5.1 (1976): 15-17.
28. Murthy, Sreerama K. "Automatic construction of decision trees from data: A multi-disciplinary survey." Data mining and knowledge discovery 2.4 (1998): 345-389.
29. Hand, David J., Heikki Mannila, and Padhraic Smyth. Principles of data mining. MIT press, 2001.
30. Deng, Houtao, George Runger, and Eugene Tuv. "Bias of importance measures for multi-valued attributes and solutions." Artificial neural networks and machine Learning–ICANN 2011 (2011): 293-300.
31. Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
32. James, Gareth, et al. An introduction to statistical learning. Vol. 112. New York: springer, 2013.
33. Ho, Tin Kam. "A data complexity analysis of comparative advantages of decision forest constructors." Pattern Analysis & Applications 5.2 (2002): 102-112.
34. Altmann, André, et al. "Permutation importance: a corrected feature importance measure." Bioinformatics 26.10 (2010): 1340-1347.
35. Deng, Houtao, George Runger, and Eugene Tuv. "Bias of importance measures for multi-valued attributes and solutions." Artificial neural networks and machine Learning–ICANN 2011 (2011): 293-300.
36. Boulesteix, Anne-Laure, et al. "Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 2.6 (2012): 493-507.
37. Kelly, Regis B. "Pathways of protein secretion in eukaryotes." Science 230 (1985): 25-33.
38. Bogsch, Erik G., et al. "An essential component of a novel bacterial protein export system with homologues in plastids and mitochondria." Journal of Biological Chemistry 273.29 (1998): 18003-18006.
39. Emanuelsson, Olof, Gunnar von Heijne, and Gisbert Schneider. "Analysis and prediction of mitochondrial targeting peptides." Methods in cell biology 65 (2001): 175-187.
40. Nickel, Walter. "Pathways of unconventional protein secretion." Current opinion in biotechnology 21.5 (2010): 621-626.
41. Bendtsen, Jannick Dyrløv, et al. "Feature-based prediction of non-classical and leaderless protein secretion." Protein Engineering Design and Selection 17.4 (2004): 349-356.
42. Drews, Jürgen. "Drug discovery: a historical perspective." Science 287.5460 (2000): 1960-1964.
43. Antoranz, Asier, et al. "Mechanism-based biomarker discovery." Drug discovery today (2017).
44. Martoglio, Bruno, and Bernhard Dobberstein. "Signal sequences: more than just greasy peptides." Trends in cell biology 8.10 (1998): 410-415.
45. Nielsen, Henrik, Søren Brunak, and Gunnar von Heijne. "Machine learning approaches for the prediction of signal peptides and other protein sorting signals." Protein engineering 12.1 (1999): 3-9.
46. Huh, Won-Ki, et al. "Global analysis of protein localization in budding yeast." Nature 425.6959 (2003): 686-691.

Πηγές εικόνων:

Εικόνα 1> Nature Reviews
Εικόνα 2. http://what-when-how.com/molecular-biology/signal-peptide-molecular-biology/
Εικόνα 3. Nature Reviews Molecular Biology
Εικόνα 4. Human Protein Atlas
Εικόνα 5. Pathways of Unconventional Secretion
Εικόνα 6. Blobel and co-workers
Εικόνα 7. Wikipedia, Artificial Neural Networks
Εικόνα 8. TMHMM:transmembrane topology prediction
Εικόνα 9. A combined transmembrane topology and signal peptide prediction method
Εικόνα 10. Wikipedia, Confusion matrix
Εικόνα 11. http://csr.ufmg.br/dinamica/dokuwiki/doku.php?id=roc_suite
Εικόνα 12. https://www.biostat.wisc.edu/~page/rocpr.pdf

Εισαγωγή

Δεδομένα πρωτεϊνών

Εκτίμηση των επιδόσεων
των προγραμμάτων πρόβλεψης

Δημιουργία αλγορίθμου πρόβλεψης

Ανάλυση των στοιχείων επαλήθευσης
της κυτταρικής τοποθεσίας μιας πρωτεΐνης

Πηγές | Βιβλιογραφία

Παράρτημα

Επίλογος | Συμπεράσματα | Περιορισμοί

Ανάλυση των λανθασμένων αποτελεσμάτων

10. Παράρτημα
Part 1. Libraries

```{r load libraries}
library(tidyverse)
library(stringr)
library(knitr)
library(SDMTools)
library(seqinr)
library(mltools)
library(PRROC)
library(glmnet)
library(boot)
library(randomForest)
library(ranger)
library(caret)
library(gplots)
```

Part 2. Loading Data

```{r Part1. Load and transform uniprot}

#In this part we load the data from uniprot and transform it only removing the proteins that we have no information.
#In df1 we have these proteins

df1 <- read_tsv("uniprot-all.tab.gz") %>% ##human reviewed##
  rename(Location = `Subcellular location [CC]`)
base <- df1
df1 <- df1 %>%
  #remove lines that have no signal sequences and no SL
  filter(!(is.na(Location) & is.na(`Signal peptide`) & is.na(Transmembrane) & is.na(Intramembrane))) %>%
  mutate(Location = str_replace_all(Location, "SUBCELLULAR LOCATION: ", ""), #erase the SUBCELLULAR string#
       Location = str_replace_all(Location, "Note=.*$", ""),  ##erase the Note:... till end#
       Location = str_replace_all(Location, "Ref\\.", "Ref"),
       Location = str_split(Location, "\\."),
       Location = map(Location, str_trim)) %>% #trim whitespace from start and end#
  unnest(Location) %>%   #unnest the vector of location#
  filter(!is.na(Location), Location != "") %>%   #remove the NA locations and blank locations#
  mutate(Location = tolower(Location),  #turn to lowercase#
       Location = str_split(Location, ";")) %>%  #split at the ;#
  unnest(Location) %>% #unnest the vector#
  mutate(Evidence = str_extract_all(Location, "eco:\\d+"), #create a new column with the evidence#
       Evidence = map(Evidence, unique),  #
       Location = str_replace_all(Location, " \\{.*\\}", "")) #erase the string inside {}#
```

Part 3. Data transformation

```{r Part 2. Continuing with the transformation}
#In this part we continue with the transformation and in df2 we have the same info as df1 without duplicates


df1 <- df1 %>%
  mutate(Membrane = grepl("membrane",Location),
       Secreted = grepl("secreted",Location),
       Extra = grepl("extracellular",Location),
       Intra = !grepl(paste(c("membrane","secreted","extracellular"),
                    collapse = "|"), Location)) %>%
  group_by(Entry) %>%
  mutate(PureSecreted = all(Secreted == TRUE) | all(Extra == TRUE),
       PureIntra = all(Intra == TRUE),
       PureMembrane = all(Membrane == TRUE)) %>%
  ungroup() %>%
  filter(PureIntra == TRUE | PureMembrane == TRUE | PureSecreted == TRUE ) %>%
  select(-Location,-Evidence) %>%
  distinct() %>%
  mutate(Domain = "blabla") %>%
  mutate(Domain = if_else(!is.na(`Signal peptide`) & (is.na(Transmembrane) & is.na(Intramembrane)) ,
                "SP only", Domain),
       Domain = if_else(!is.na(`Signal peptide`) & (!is.na(Transmembrane) | !is.na(Intramembrane)),
                "SP and Mem.",Domain),
       Domain = if_else(is.na(`Signal peptide`) & (!is.na(Transmembrane) | !is.na(Intramembrane)),
                "Mem. only",Domain),
       Domain = if_else(is.na(`Signal peptide`) & is.na(Transmembrane) & is.na(Intramembrane),
                "No domain",Domain),
       Location = "blabla",
       Location = if_else(PureSecreted == TRUE, "Secreted", Location),
       Location = if_else(PureMembrane == TRUE, "Membrane", Location),
       Location = if_else(PureIntra == TRUE, "Intracellular", Location),
       Domain = as.factor(Domain),
       Location = as.factor(Location)) %>%
  mutate(hasSP = ifelse((Domain == 'SP only' ) | (Domain == 'SP and Mem.'), yes = 1, no = 0),Domain ) %>%
  distinct()

df2 <- df1[!duplicated(df1$Entry),]
df2 <- df2 %>%
  mutate(hasTM = if_else(is.na(Transmembrane), 0 , 1),
       Secreted = if_else((Location == 'Intracellular' | Location == 'Membrane'), FALSE, TRUE))



```

Part 4. Data Content

```{r Part 3. Presenting the info on the df2 dataset}

#This is just output and loading the output of the programs that follow
##################################
load("~/Documents/SignalP/signalp1.Rda")
load("~/Documents/SignalP/tmhmm1.Rda")
load("~/Documents/SignalP/phobius1.Rda")
load("~/Documents/SignalP/wolf11.Rda")
##################################

t1 <- table(df2$Secreted,df2$hasSP)
rownames(t1) <- c("Not secreted","Secreted")
t2 <- table(df2$Secreted,df2$hasTM)
rownames(t2) <- c("Not secreted","Secreted")
t3 <- table(df2$Secreted,df2$Domain)
rownames(t3) <- c("Not secreted","Secreted")
t4 <- table(df2$Location,df2$Domain)
kable(t1,caption = "Secreted ~ Signal Peptide",col.names = c("No SP", "SP"),align = "c", row.names = TRUE)
kable(t2,caption = "Secreted ~ TM domain",col.names = c("No TM", "TM"),align = "c")
kable(t3,caption = "Secreted ~ Signal Peptide + TM domain",col.names = c("TM + NO SP", "NO TM + NO SP","TM
+ SP", "NO TM + SP"), align = "c")
kable(t4,caption = "Location ~ Signal Peptide + TM domain",col.names = c("TM + NO SP", "NO TM + NO SP","TM
+ SP", "NO TM + SP"), align = "c")
```

Part 5. SignalP

```{r Part 4. SignalP, eval = FALSE}

#this part calls signalp on the df2 dataset we have eval false because it takes a lot of time to run

call_signalp <- function(input,output) {

library(tidyverse)


wd <- file.path(getwd(),'signalp-4.1')
in_file <- file.path(wd,'test',input)
out_file <- file.path(wd,'test',output)
system(sprintf('%s/signalp -t euk -f short %s > %s', wd, in_file, out_file))

signalp_out <- read_table(out_file, col_names = TRUE , skip = 1) %>%
  #separate(`# name`, c('SP','UniProtID','ProtName')) %>%
  select(`# name`,D)
return(signalp_out)

}
```

Part 6. SignalP evaluation

```{r Part 5 . SignalP evaluation}

#we evaluate the performance of signalp

confsig <- confusion.matrix(df2$hasSP,signalp_out$D,threshold = 0.5)
accsig <- accuracy(df2$hasSP,signalp_out$D,threshold = 0.5)

precsig <- confsig[2,2]/sum(confsig[2,])
mccSigP <- mcc(TP = confsig[2,2], FP = confsig[2,1], TN = confsig[1,1], FN = confsig[1,2])
perfsig <- cbind(accsig,precsig,mccSigP)
confsig <- as.table(confsig)
kable(confsig, caption = "SignalP confusion matrix", align = "c")
kable(perfsig,caption = "SignalP Performance", align = "c")
fg <- signalp_out$D[df2$hasSP == 1]
bg <- signalp_out$D[df2$hasSP == 0]
pr <- pr.curve(scores.class0 = fg, scores.class1 = bg, curve = T)
plot(pr)

```

Part 7. TMHMM

```
``` {r Part 6. tmhmm, eval = FALSE}

#we call tmhmm on the df2 dataset

call_tmhmm <- function(input,output) {

library(tidyverse)
wd <- file.path(getwd(),'tmhmm-2.0c/bin')

in_file <- file.path(wd,'test',input)
out_file <- file.path(wd,'test',output)

system(sprintf('%s/tmhmm  %s > %s', wd, in_file, out_file))

tmhmm_out <- read_table2(out_file, col_names = FALSE, skip = 0 , comment = "-" )
colnames(tmhmm_out) <- c("name","length","v3","v4","helices","topology")

tmhmm_out <- tmhmm_out %>%
  select(name,helices) %>%
  mutate(helices = str_replace_all(helices,"PredHel=",""),
       TM = if_else(helices == 0, 0 , 1, ))

return(tmhmm_out)
}

write.fasta(as.list(df2$Sequence) , as.string = TRUE, df2$Entry, "/home/chris/Documents/SignalP/tmhmm-
2.0c/bin/test/tmhmmtest1")
tmhmm_out <- call_tmhmm("tmhmmtest1","tmhmmtestout1")

tmhmm_out <- tmhmm_out %>%
select(name,TM)
```

Part 8. TMHMM evaluation

```
```{r Part 7. tmhmm evaluation}

#we evaluate the performance of tmhmm

conftm <- confusion.matrix(df2$hasTM,tmhmm_out$TM,threshold = 0.5)
acctm <- accuracy(df2$hasTM,tmhmm_out$TM,threshold = 0.5)
prectm <- conftm[2,2]/sum(conftm[2,])
mcctm <- mcc(TP = conftm[2,2], FP = conftm[2,1], TN = conftm[1,1], FN = conftm[1,2])
perftm <- cbind(acctm,prectm,mcctm)
conftm <- as.table(conftm)
kable(conftm, caption = "TMHMM confusion matrix", align = "c")
```

Part 9. Phobius

```{r Part 8. Phobius, eval = FALSE}

#call phobius on df2

call_phobius <- function(input,output) {
wd <- file.path(getwd(),'phobius')

in_file <- file.path(wd,input)
out_file <- file.path(wd,output)

system(sprintf('perl /home/chris/Documents/SignalP/phobius/phobius.pl -short  %s > %s', in_file, out_file))

phobius_out <- read_table2(out_file, col_names = FALSE, skip = 0 , comment = "-" )

return(phobius_out)
}
write.fasta(as.list(df2$Sequence)  ,  as.string  =  TRUE,  df2$Entry, "/home/chris/Documents/SignalP/phobius/phobiustest")
phobius_out <- call_phobius("phobiustest","phobiustestout")

phobius_out <- read_table2("/home/chris/Documents/SignalP/phobius/phobiustestout", col_names = TRUE, skip = 0 , comment = "-" , guess_max = Inf ) %>%
  select(SEQENCE,ID,TM)
colnames(phobius_out) <- c("entry","TM","SP")

phobius_out <- phobius_out %>%
  mutate(phobius_tm = if_else(TM != 0, 1 , 0)) %>%
  mutate(phobius_sp = if_else(SP == "Y", 1, 0)) %>%
  select(entry,phobius_sp,phobius_tm)
```

Part 10. Phobius evaluation

```{r Part 9. phobius evaluation}

confphtm <- confusion.matrix(df2$hasTM,phobius_out$phobius_tm,threshold = 0.5)
accphtm <- accuracy(df2$hasTM,phobius_out$phobius_tm,threshold = 0.5)
precphtm <- confphtm[2,2]/sum(confphtm[2,])
mccphtm <- mcc(TP = confphtm[2,2], FP = confphtm[2,1], TN = confphtm[1,1], FN = confphtm[1,2])
perfphtm <- cbind(accphtm,precphtm,mccphtm)
confphtm <- as.table(confphtm)
kable(confphtm, caption = "Phobius TM confusion matrix", align = "c")
kable(perfphtm,caption = "Phobius TM Performance", align = "c")

confphsp <- confusion.matrix(df2$hasSP,phobius_out$phobius_sp,threshold = 0.5)
accphsp <- accuracy(df2$hasSP,phobius_out$phobius_sp,threshold = 0.5)
precphsp <- confphsp[2,2]/sum(confphsp[2,])
mccphsp <- mcc(TP = confphsp[2,2], FP = confphsp[2,1], TN = confphsp[1,1], FN = confphsp[1,2])
perfphsp <- cbind(accphsp,precphsp,mccphsp)
confphsp <- as.table(confphsp)
kable(confphsp, caption = "Phobius SP confusion matrix", align = "c")
kable(perfphsp,caption = "Phobius SP Performance", align = "c")

```

Part 11. WolfPSORT

```{r Part 10. wolfpsort, eval = FALSE}


call_wolf <- function(input,output) {

  library(tidyverse)


  wd <- file.path(getwd(),'WoLFPSort-master/bin')
  in_file <- file.path(wd,input)
  out_file <- file.path(wd,output)
  system(sprintf('%s/runWolfPsortSummary animal <%s >%s', wd, in_file, out_file))

  wolf_out <- read_table(out_file, col_names = FALSE , skip = 0)
    #separate(`# name`, c('SP','UniProtID','ProtName'))

  return(wolf_out)

}

write.fasta(as.list(df2$Sequence) , as.string = TRUE, df2$Entry, "/home/chris/Documents/SignalP/WoLFP-Sort-master/bin/wolftest")
wolf_out <- call_wolf("wolftest","wolftestout")

wolf_out <- read_table("WoLFPSort-master/bin/wolftestout", skip = 0)
colnames(wolf_out) <- "x1"

wolf_out <- wolf_out %>%
  mutate(location = word(string = x1,2,3),
        location = str_replace_all(location,",,,"),
        x1 = word(x1,1,1),
        power1 = word(location,2,2),
        location1 = word(location,1,1),
        power2 = power1)

wolf_out <- wolf_out %>%
  select(x1,location1,power2)

colnames(wolf_out) <- c("entry","location","knn")

wolf_out <- wolf_out %>%
  mutate(Secreted = grepl("extr", location),
        Secreted = as.numeric(Secreted))
df2 <- df2 %>%
  mutate(Secreted = as.numeric(Secreted))
wolf_out <- as.data.frame(wolf_out)


```

Part 12. WolfPSORT evaluation

```{r Part 11. wolfpsort evaluation}

test <- semi_join(wolf_out,df2, by = c("entry" = "Entry"))
test1 <- semi_join(df2,wolf_out, by = c("Entry" = "entry"))
confwolf <- confusion.matrix(test1$Secreted,test$Secreted,threshold = 0.5)
accwolf <- accuracy(test1$Secreted,test$Secreted,threshold = 0.5)
precwolf <- confwolf[2,2]/sum(confwolf[2,])
mccwolf <- mcc(TP = confwolf[2,2], FP = confwolf[2,1], TN = confwolf[1,1], FN = confwolf[1,2])
perfwolf <- cbind(accwolf,precwolf,mccwolf)
confwolf <- as.table(confwolf)
kable(confwolf, caption = "WolfPSORT confusion matrix", align = "c")
kable(perfwolf,caption = "WolfPSORT Performance", align = "c")


```

Part 13. Training set

```{r Part 12. Training dataset}

#we merge df2 with the output of the programs in tt and fix the names

tt <- full_join(df2,wolf_out, by = c("Entry" = "entry"))
names(tt)[names(tt) == "location"] <- "WolfLoc"
names(tt)[names(tt) == "Secreted.y"] <- "WolfSec"
phobius_out <- as.data.frame(phobius_out)
tt <- full_join(tt,phobius_out, by = c("Entry" = "entry"))
signalp_out <- as.data.frame(signalp_out)
tt <- full_join(tt,signalp_out, by = c("Entry" = "# name"))
names(tt)[names(tt) == "D"] <- "SigP"
tmhmm_out <- as.data.frame(tmhmm_out)
tt <- full_join(tt,tmhmm_out, by = c("Entry" = "name"))
names(tt)[names(tt) == "TM"] <- "tmhmm"
tt <- tt %>%
  select(Entry,Sequence,Secreted.x,WolfSec,knn,phobius_sp,phobius_tm,SigP,tmhmm)
tt <- tt[c("Entry","Sequence","SigP","tmhmm","phobius_sp","phobius_tm","WolfSec","knn","Secreted.x")]
names(tt)[names(tt) == "Secreted.x"] <- "Secreted"
names(tt)[names(tt) == "knn"] <- "wolfknn"
```

Part 14. Random Forest

```{r Part 15. Random forest with parameter tuning}

#here we train a random forest model using mtry as the tuning parameter
aa <- aa %>%
  mutate(Secreted = ifelse(Secreted == TRUE , 1 , 0),
        Secreted = as.factor(Secreted))
seeds <- set.seed(56)
caretrg <- train(
    Secreted ~ .
   ,data = aa
   ,method = "ranger"
   ,metric = "Kappa"
   ,trControl = trainControl(method = "cv", number = 10, allowParallel = TRUE, verbose = FALSE, seeds = seeds)
   ,tuneGrid = expand.grid(mtry = c(3,4,5), splitrule = c("gini","extratrees"))
   ,importance = 'impurity'
)


predcarrg <- predict(caretrg,aa)
confmodel5 <- confusion.matrix(aa$Secreted,predcarrg,threshold = 0.5)
accmodel5 <- accuracy(aa$Secreted,predcarrg,threshold = 0.5)
precmodel5 <- confmodel5[2,2]/sum(confmodel5[2,])
mccmodel5 <- mcc(TP = confmodel5[2,2], FP = confmodel5[2,1], TN = confmodel5[1,1], FN = confmodel5[1,2])
perfmodel5 <- cbind(accmodel5,precmodel5,mccmodel5)
confmodel5 <- as.table(confmodel5)
perfmodel5 <- perfmodel5[-7]
kable(confmodel5, caption = "Ranger forest w/ caret w/parameter tuning confusion matrix", align = "c")
kable(perfmodel5,caption = "Ranger forest  w/ caret w/parameter tuning Performance", align = "c")
caretrg
kable(caretrg$results)
varImp(caretrg)
```

Part 15. SignalP and TMHMM test

```{r Part 15.5 testing SignalP and tmhmm combined power}

bb <- aa
bb <- bb %>%
  mutate(Secreted2 = if_else(SigP > 0.5 & tmhmm == 0 , 1 , 0))

confmodel55 <- confusion.matrix(aa$Secreted,bb$Secreted2,threshold = 0.5)
accmodel55 <- accuracy(aa$Secreted,bb$Secreted2,threshold = 0.5)
precmodel55 <- confmodel55[2,2]/sum(confmodel55[2,])
mccmodel55 <- mcc(TP = confmodel55[2,2], FP = confmodel55[2,1], TN = confmodel55[1,1], FN = confmodel55[1,2])
perfmodel55 <- cbind(accmodel55,precmodel55,mccmodel55)
confmodel55 <- as.table(confmodel55)
kable(confmodel55, caption = "SignalP and TMHMM combination", align = "c")
kable(perfmodel55,caption = "SignalP and TMHMM combination", align = "c")
```

Part 16. Loading Data for ECO analysis

```{r Part 16. Load uni}
#load from uniprot in the main dataframe this is almost exactly the same as with the first df but #with some extra changes
main <- read_tsv("uniprot-all.tab.gz") %>% ##human reviewed##
  rename(Location = `Subcellular location [CC]`)

main <- main %>%
  #remove lines that have no signal sequences and no SL
  filter(!(is.na(Location) & is.na(`Signal peptide`) & is.na(Transmembrane) & is.na(Intramembrane))) %>%
  mutate(Location = str_replace_all(Location, "SUBCELLULAR LOCATION: ", ""), #erase the SUBCELLULAR string#
       Location = str_replace_all(Location, "Note=.*$", ""),  ##erase the Note:... till end#
       Location = str_replace_all(Location, "Ref\\.", "Ref"),
       Location = str_split(Location, "\\."),
       Location = map(Location, str_trim)) %>% #trim whitespace from start and end#
  unnest(Location) %>%   #unnest the vector of location#
  filter(!is.na(Location), Location != "") %>%   #remove the NA locations and blank locations#
  mutate(Location = tolower(Location),  #turn to lowercase#
       Location = str_split(Location, ";")) %>%  #split at the ;#
  unnest(Location) %>% #unnest the vector#
  mutate(Evidence = str_extract_all(Location, "eco:\\d+"), #create a new column with the evidence#
       Evidence = map(Evidence, unique),  #
       Location = str_replace_all(Location, " \\{.*\\}", "")) #erase the string inside {}#
```

```
main$Evidence[map_int(main$Evidence, length) == 0] = c(" ")  # hackeria gia na min svisei grammes
main <- main %>%
  unnest(Evidence) %>% #unnest the different evidence from the vector#
  mutate(Evidence = if_else(Evidence == " ", NA_character_, Evidence), #put NA on evidence if its nothing#
       Location = str_split(Location, ",")) %>% #split at the comma#
  unnest(Location) %>%                    #unnest the locations#
  mutate(Location = str_trim(Location)) %>%  #remove the whitespaces#
  filter(Location != "")   #remove the no locations that came from splitting and unnesting at ","#

main <- main %>%
  mutate(Domain = "blabla") %>%
  mutate(Domain = if_else(!is.na(`Signal peptide`) & (is.na(Transmembrane) & is.na(Intramembrane)) ,
                 "SP only", Domain),
       Domain = if_else(!is.na(`Signal peptide`) & (!is.na(Transmembrane) | !is.na(Intramembrane)),
                 "SP and Mem.",Domain),
       Domain = if_else(is.na(`Signal peptide`) & (!is.na(Transmembrane) | !is.na(Intramembrane)),
                 "Mem. only",Domain),
       Domain = if_else(is.na(`Signal peptide`) & is.na(Transmembrane) & is.na(Intramembrane),
                 "No domain",Domain),

       Domain = as.factor(Domain)) %>%
   mutate(hasSP = ifelse((Domain == 'SP only' ) | (Domain == 'SP and Mem.'), yes = 1, no = 0),Domain ) %>%
  mutate(hasTM = if_else(is.na(Transmembrane), 0 , 1))

#this is changed that proteins with no evidence are filtered out###
main <- main %>%
  group_by(Entry) %>%
  filter(!all(is.na(Evidence))) %>%  #filter out all the proteins that dont provide any evidence#
  filter(!is.na(Evidence)) %>%
  ungroup()

#############this is also changed the way that we create the secreted column#####################
main <- main %>%
  mutate(Secreted = grepl(paste(c("secreted","extracellular"),
                    collapse = "|"), Location)) %>%
  select(Entry,Location,Evidence,Domain,hasSP,hasTM,Secreted)
##############################################################
```

Part 17. Data transformation

```{r Part 17. split into 5 new data frames for each eco}

eco250 <- main[grepl("250",main$Evidence),]
eco255 <- main[grepl("255",main$Evidence),]
eco269 <- main[grepl("269",main$Evidence),]
eco305 <- main[grepl("305",main$Evidence),]
eco303 <- main[grepl("303",main$Evidence),]

eco250 <- eco250 %>%
  group_by(Entry) %>%
  filter(all(Secreted == TRUE) | all(Secreted == FALSE)) %>%   ###keep proteis that have clean secreted anno-
tation#
  mutate(Location = paste(Location,collapse = ", ")) %>%
  ungroup() %>%
  distinct()
eco250 <- eco250[!duplicated(eco250$Entry),]

eco255 <- eco255 %>%
  group_by(Entry) %>%
  filter(all(Secreted == TRUE) | all(Secreted == FALSE)) %>%
  mutate(Location = paste(Location,collapse = ", ")) %>%
  ungroup() %>%
  distinct()
eco255 <- eco255[!duplicated(eco255$Entry),]

eco269 <- eco269 %>%
  group_by(Entry) %>%
  filter(all(Secreted == TRUE) | all(Secreted == FALSE)) %>%
  mutate(Location = paste(Location,collapse = ", ")) %>%
  ungroup() %>%
  distinct()
eco269 <- eco269[!duplicated(eco269$Entry),]

eco303 <- eco303 %>%
  group_by(Entry) %>%
  filter(all(Secreted == TRUE) | all(Secreted == FALSE)) %>%
  mutate(Location = paste(Location,collapse = ", ")) %>%
  ungroup() %>%
  distinct()
eco303 <- eco303[!duplicated(eco303$Entry),]


  distinct()
eco305 <- eco305[!duplicated(eco305$Entry),]

#show the overlap of entries for the data sets
```

```
eco305 <- eco305 %>%
  group_by(Entry) %>%
  filter(all(Secreted == TRUE) | all(Secreted == FALSE)) %>%
  mutate(Location = paste(Location,collapse = ", ")) %>%
  ungroup() %>%
  distinct()
eco305 <- eco305[!duplicated(eco305$Entry),]

#show the overlap of entries for the data sets
venn(list(eco250 = eco250$Entry, eco255 = eco255$Entry, eco269 = eco269$Entry, eco303 = eco303$Entry,
eco305 = eco305$Entry))


```
```
```

Part 18. ECO analysis

```{r Part 18. Eco250 vs Eco255}

#first we will check eco250 and eco255 for conflicts
venn(list(eco250 = eco250$Entry, eco255 = eco255$Entry)) #venn
a250 <- which(eco250$Entry %in% intersect(eco250$Entry,eco255$Entry)) #intersect
a255 <- which(eco255$Entry %in% intersect(eco250$Entry,eco255$Entry)) #intersect
dis0 <- arrange(eco250[a250,],Entry) #arrange the common ones
dis0b <- arrange(eco255[a255,],Entry) #arrange the common ones
c0 <- dis0$Secreted != dis0b$Secreted #check for conflicting locations
# no conflicting annotations
#can bind with ease

###bind
eco250255 <- rbind(eco250,eco255) %>%
  group_by(Entry) %>%
  mutate(Location = paste(Location,collapse = " / ")) %>%
  mutate(Evidence = paste(Evidence,collapse = " / ")) %>%
  ungroup() %>%
  distinct()
#/bind

```

```{r Part 19. Eco269 vs Eco303}

#check 269 vs 303
venn(list(eco269 = eco269$Entry, eco303 = eco303$Entry))
a3 <- which(eco269$Entry %in% intersect(eco269$Entry,eco303$Entry)) #inter
a4 <- which(eco303$Entry %in% intersect(eco269$Entry,eco303$Entry)) #inter
#check for conflicts
dis3 <- arrange(eco269[a3,],Entry)
dis4 <- arrange(eco303[a4,],Entry)
c2 <- dis3$Secreted != dis4$Secreted

#no conflicts between eco 269 vs 303
#can bind them and then recheck but keep the respective evidence
####bind
eco269303 <- rbind(eco269,eco303) %>%
  group_by(Entry) %>%
  mutate(Location = paste(Location,collapse = " / ")) %>%
  mutate(Evidence = paste(Evidence,collapse = " / ")) %>%
  ungroup() %>%
  distinct()
#/bind
```

```{r Part 20. Eco269303 and Eco305}

#269303 vs 305 checking
venn(list(eco269303 = eco269303$Entry, eco305 = eco305$Entry))
a5 <- which(eco269303$Entry %in% intersect(eco269303$Entry,eco305$Entry)) #inter
a6 <- which(eco305$Entry %in% intersect(eco269303$Entry,eco305$Entry)) #inter
#check for conflicts there are 27 conflicts
dis5 <- arrange(eco269303[a5,],Entry)
dis6 <- arrange(eco305[a6,],Entry)
c3 <- dis5$Secreted != dis6$Secreted

#we will create a new df for the conflicts
conflict1a <- dis5[c3,]
conflict1b <- dis6[c3,]
conflict1 <- rbind(conflict1a,conflict1b) %>%
  arrange(Entry) %>%
  group_by(Entry) %>%
  mutate(Evidence = paste(Evidence,collapse = " / ")) %>%
  mutate(Secreted = paste(Secreted,collapse = " / ")) %>%
  mutate(Location = paste(Location,collapse = " / ")) %>%
  ungroup() %>%
  distinct()

#we will now bind and remove the conflicting proteins
####bind
eco269303305 <- rbind(eco269303,eco305)
a7 <- which(eco269303305$Entry %in% intersect(eco269303305$Entry,conflict1$Entry))
eco269303305 <- eco269303305[-a7,] %>% #remove
  group_by(Entry) %>%
  mutate(Location = paste(Location,collapse = " / ")) %>%
  mutate(Evidence = paste(Evidence,collapse = " / ")) %>%
  ungroup() %>%
  distinct()
#/bind
```

```{r Part 21. Final binding and conflicts}

#the final dataset will be in ecoall and the conflicts will be in conflicts

#we now have to do the final binding 250255 & 269303305
venn(list(eco269303305 = eco269303305$Entry, eco250255 = eco250255$Entry))
a8 <- which(eco269303305$Entry %in% intersect(eco269303305$Entry,eco250255$Entry))
a9 <- which(eco250255$Entry %in% intersect(eco269303305$Entry,eco250255$Entry))
#check for conflicting locations
dis7 <- arrange(eco269303305[a8,],Entry)
dis8 <- arrange(eco250255[a9,],Entry)
c4 <- dis7$Secreted != dis8$Secreted
#there are 45 conflicts
```

```
#create a new df for the conflicts
conflict2a <- dis7[c4,]
conflict2b <- dis8[c4,]
conflict2 <- rbind(conflict2a,conflict2b) %>%
  arrange(Entry) %>%
  group_by(Entry) %>%
  mutate(Evidence = paste(Evidence,collapse = " / ")) %>%
  mutate(Secreted = paste(Secreted,collapse = " / ")) %>%
  mutate(Location = paste(Location,collapse = " / ")) %>%
  ungroup() %>%
  distinct()

#bind all the conflicts together
venn(list(conf2 = conflict2$Entry, conf1 = conflict1$Entry))
conflicts <- rbind(conflict1,conflict2) %>%
  arrange(Entry) %>%
  group_by(Entry) %>%
  mutate(Evidence = paste(Evidence,collapse = " // ")) %>%
  mutate(Secreted = paste(Secreted,collapse = " // ")) %>%
  mutate(Location = paste(Location,collapse = " // ")) %>%
  ungroup() %>%
  distinct()

#bind
ecoall <- rbind(eco269303305,eco250255)
a10 <- which(ecoall$Entry %in% intersect(ecoall$Entry,conflict2$Entry))
ecoall <- ecoall[-a10,] %>% #remove
  group_by(Entry) %>%
  mutate(Location = paste(Location,collapse = " / ")) %>%
  mutate(Evidence = paste(Evidence,collapse = " / ")) %>%
  ungroup() %>%
  distinct()
#/bind
#ecoall is my final dataset with no conflicts
kable(head(conflicts))
cnf <- which(base$Entry %in% intersect(base$Entry,conflicts$Entry))

```
```
```

Part 19. Retraining Random Forest for ECO analysis

```{r Part 22. Retrain forest with the ecoall df}

#tt has the features from the previous training set, we will subset from that to get the feature values for the ecoall dataset
venn(list(eco = ecoall$Entry, tt = tt$Entry))
#which in the eco are not in my tts
a <- which(ecoall$Entry %in% intersect(tt$Entry,ecoall$Entry))
b <- which(tt$Entry %in% intersect(tt$Entry,ecoall$Entry))
thediff <- ecoall[-a,]
#i will have to call my feature prediction programs on those
am <- which(base$Entry %in% intersect(base$Entry,thediff$Entry))
```

```{r Part 23. Calling feature prediction programs, eval = FALSE}
#sigp
write.fasta(as.list(base$Sequence[am]) , as.string = TRUE, base$Entry[am], "/home/chris/Documents/SignalP/signalp-4.1/test/signalptt")
signalp_out2 <- call_signalp("signalptt","ttout")
colnames(signalp_out2) <- c("Entry","SigP")
#tmhmm
write.fasta(as.list(base$Sequence[am]) , as.string = TRUE, base$Entry[am], "/home/chris/Documents/SignalP/tmhmm-2.0c/bin/test/tmhmmtt1")
tmhmm_out2 <- call_tmhmm("tmhmmtt1","tmhmmtestt1")
tmhmm_out2 <- tmhmm_out2 %>%
select(name,TM)
colnames(tmhmm_out2) <- c("Entry","tmhmm")
#phobius
write.fasta(as.list(base$Sequence[am]) , as.string = TRUE, base$Entry[am], "/home/chris/Documents/SignalP/phobius/phobiustt")
phobius_out2 <- call_phobius("phobiustt","phobiustestt")

phobius_out2 <- read_table2("/home/chris/Documents/SignalP/phobius/phobiustestt", col_names = TRUE, skip = 0 , comment = "-" , guess_max = Inf ) %>%
  select(SEQENCE,ID,TM)
colnames(phobius_out2) <- c("Entry","TM","SP")

phobius_out2 <- phobius_out2 %>%
  mutate(phobius_tm = if_else(TM != 0, 1 , 0)) %>%
  mutate(phobius_sp = if_else(SP == "Y", 1, 0)) %>%
  select(Entry,phobius_sp,phobius_tm)
```

```
#wolf
write.fasta(as.list(base$Sequence[am]) , as.string = TRUE, base$Entry[am], "/home/chris/Documents/SignalP/
WoLFPSort-master/bin/wolftt")
wolf_out2 <- call_wolf("wolftt","wolftestt")

wolf_out2 <- read_table("WoLFPSort-master/bin/wolftestt",  skip = 0)
colnames(wolf_out2) <- "x1"

wolf_out2 <- wolf_out2 %>%
  mutate(location = word(string = x1,2,3),
       location = str_replace_all(location,",,""),
       x1 = word(x1,1,1),
       power1 = word(location,2,2),
       location1 = word(location,1,1),
       power2 = power1)

wolf_out2 <- wolf_out2 %>%
  select(x1,location1,power2)

colnames(wolf_out2) <- c("Entry","location","knn")

wolf_out2 <- wolf_out2 %>%
  mutate(Secreted = grepl("extr", location),
       Secreted = as.numeric(Secreted))

wolf_out2 <- as.data.frame(wolf_out2) %>%
  select(Entry,Secreted)
colnames(wolf_out2) <- c("Entry","WolfSec")
```

```{r Part 24. New training set from ecoall with no conflicts}
#tteco now has the info as tt
#aan will be the training set

######load the output of the prediction programs
load("~/Documents/SignalP/signalp2.Rda")
load("~/Documents/SignalP/tmhmm2.Rda")
load("~/Documents/SignalP/phobius2.Rda")
load("~/Documents/SignalP/wolf2.Rda")
######
#new training set
ttnew <- tt[b,]
ttdiff <- inner_join(signalp_out2,tmhmm_out2, by = "Entry")
ttdiff <- inner_join(ttdiff,phobius_out2, by = "Entry")
ttdiff <- inner_join(ttdiff,wolf_out2,by = "Entry")
ttdiff <- inner_join(ttdiff,thediff, by = "Entry")
ttnew2 <- ttdiff %>%
  select(Entry,SigP,tmhmm,phobius_sp,phobius_tm,WolfSec,Secreted)
```

```
ttnew <- rbind(ttnew,ttnew2)

ttnew <- ttnew %>%
  mutate(WolfSec = ifelse(is.na(WolfSec),0,WolfSec))
tteco <- ttnew
aan <- ttnew %>%
  select(-Entry)
```

Part 20. Conflict analysis

```{r Part 26. Call feature prediction programs on conflicts, eval = FALSE}

#sigp
write.fasta(as.list(base$Sequence[cnf]) , as.string = TRUE, base$Entry[cnf], "/home/chris/Documents/SignalP/signalp-4.1/test/signalpcnf")
signalp_out3 <- call_signalp("signalpcnf","cnfout")
colnames(signalp_out3) <- c("Entry","SigP")
#tmhmm
write.fasta(as.list(base$Sequence[cnf]) , as.string = TRUE, base$Entry[cnf], "/home/chris/Documents/SignalP/tmhmm-2.0c/bin/test/tmhmmcnf")
tmhmm_out3 <- call_tmhmm("tmhmmcnf","tmhmmtestcnf")
tmhmm_out3 <- tmhmm_out3 %>%
select(name,TM)
colnames(tmhmm_out3) <- c("Entry","tmhmm")
#phobius
write.fasta(as.list(base$Sequence[cnf]) , as.string = TRUE, base$Entry[cnf], "/home/chris/Documents/SignalP/phobius/phobiuscnf")
phobius_out3 <- call_phobius("phobiuscnf","phobiustestcnf")

phobius_out3 <- read_table2("/home/chris/Documents/SignalP/phobius/phobiustestcnf", col_names = TRUE, skip = 0 , comment = "-" , guess_max = Inf ) %>%
  select(SEQENCE,ID,TM)
colnames(phobius_out3) <- c("Entry","TM","SP")

phobius_out3 <- phobius_out3 %>%
  mutate(phobius_tm = if_else(TM != 0, 1 , 0)) %>%
  mutate(phobius_sp = if_else(SP == "Y", 1, 0)) %>%
  select(Entry,phobius_sp,phobius_tm)

#wolf
write.fasta(as.list(base$Sequence[cnf]) , as.string = TRUE, base$Entry[cnf], "/home/chris/Documents/SignalP/WoLFPSort-master/bin/wolfcnf")
wolf_out3 <- call_wolf("wolfcnf","wolftestcnf")

wolf_out3 <- read_table("WoLFPSort-master/bin/wolftestcnf",  skip = 0)
colnames(wolf_out3) <- "x1"

wolf_out3 <- wolf_out3 %>%
  mutate(location = word(string = x1,2,3),
       location = str_replace_all(location,",,""),
       x1 = word(x1,1,1),
       power1 = word(location,2,2),
       location1 = word(location,1,1),
       power2 = power1)

wolf_out3 <- wolf_out3 %>%
```

```
colnames(wolf_out3) <- c("Entry","location","knn")

wolf_out3 <- wolf_out3 %>%
  mutate(Secreted = grepl("extr", location),
       Secreted = as.numeric(Secreted))

wolf_out3 <- as.data.frame(wolf_out3) %>%
  select(Entry,Secreted)
colnames(wolf_out3) <- c("Entry","WolfSec")

ttcnf <- inner_join(signalp_out3,tmhmm_out3, by = "Entry")
ttcnf <- inner_join(ttcnf,phobius_out3, by = "Entry")
ttcnf <- inner_join(ttcnf,wolf_out3,by = "Entry")
ttcnf <- ttcnf %>%
  mutate(Predictions = predict(caretrg2,newdata = ttcnf[,2:6]))
conflicts <- conflicts %>%
  mutate(Prediction = ttcnf$Predictions)
```

Part 21. False result analysis

```{r}
predictions2 <- predict(caretrg2,newdata = aan)
ecoall2 <- ecoall %>%
  mutate(predictions = predictions2)
ecoall2 <- cbind(ecoall2,aan$Secreted)
wrong1FP <- ecoall2[ecoall2$`aan$Secreted` != ecoall2$predictions & ecoall2$predictions == 1,]
wrong1FN <- ecoall2[ecoall2$`aan$Secreted` != ecoall2$predictions & ecoall2$predictions == 0,]

sc <- which(main$Entry %in% intersect(main$Entry,FPsSCpath$Entry))
scc <- which(main$Entry %in% intersect(main$Entry,corcls$Entry))
SCpathwayc <- main[scc,] %>%
  select(-Evidence) %>%
  filter(hasTM == 0)
SCpathway <- main[sc,] %>%
  select(-Evidence)
kable(head(SCpathway))
incorcls <- incorcls[,!duplicated(colnames(incorcls))]
kable(head(incorcls))
kable(head(SCpathwayc))
cci <- which(corcls$Entry %in% intersect(corcls$Entry,SCpathwayc$Entry))
corcls2 <- corcls[cci,]
corcls2 <- corcls2[,!duplicated(colnames(corcls2))]
kable(head(corcls2))
```

```{r}
FNnoSP <- FN[FN$hasSP == 0,]
fnc <- which(main$Entry %in% intersect(main$Entry,FNnoSP$Entry))
FNnoSP2 <- main[fnc,] %>%
  select(-Evidence)
kable(head(FNnoSP2))
fncc <- which(tt$Entry %in% intersect(tt$Entry,FNnoSP$Entry))
FNnoSPvar <- tt[fncc,]
kable(head(FNnoSPvar))
seqfn <- which(base$Entry %in% intersect(base$Entry,FNnoSP2$Entry))
write.fasta(as.list(base$Sequence[seqfn]) , as.string = TRUE, base$Entry[seqfn], "/home/chris/Documents/SignalP/fnnosp")
intra <- tt[tt$Secreted == FALSE,]
intra100 <- intra[1:50,]
in50 <- which(base$Entry %in% intersect(base$Entry,intra100$Entry))
write.fasta(as.list(base$Sequence[in50]) , as.string = TRUE, base$Entry[in50], "/home/chris/Documents/SignalP/intra50")
```

```{r play}

play <- cbind(tt$Entry,aa$Secreted,predcarrg)
play <- as.data.frame(play)
play2 <- cbind(tt,aa,predcarrg)
play2 <- as.data.frame(play2)
x <- play$V2 == 2 & play$predcarrg == 1
y <- play$V2 == 1 & play$predcarrg == 2
yy <-  play$V2 == 1 & play$predcarrg == 2 & df2$hasSP == 1
FN <- df2[x,] #the false negatives, they are secreted but were predicted not secreted
FP <- df2[y,]
FPSP <- FP[FP$hasSP == TRUE,]
weird1 <- FP$Entry
weird1 <- as.data.frame(weird1)
weirdSP <- FPSP$Entry
weirdSP <- as.data.frame(weirdSP)
whyweirdSP <- df2[yy,]
write.csv(x = weird1,file = "weird1",row.names = FALSE,col.names = TRUE,quote = FALSE)
write.csv(x = weirdSP,file = "FPsWithSPS",row.names = FALSE,col.names = TRUE,quote = FALSE)
#for my false positives
df4  <-  read_tsv("uniprot--P22304+O14792+P35475+Q8IWB1+Q7Z4H8+Q99538+P38571+Q14696+Q02083
+P17050--.tab.gz") %>% ##human reviewed##
  rename(Location = `Subcellular location [CC]`) %>%
  mutate(Location = str_replace_all(Location, "SUBCELLULAR LOCATION: ", ""), #erase the SUBCELLULAR string#
       Location = str_replace_all(Location, "Note=.*$", ""),  ##erase the Note:... till end#
       Location = str_replace_all(Location, "Ref\\.", "Ref"),
       Location = str_split(Location, "\\."),
       Location = map(Location, str_trim)) %>% #trim whitespace from start and end#
  unnest(Location) %>%   #unnest the vector of location#
  filter(!is.na(Location), Location != "") %>%   #remove the NA locations and blank locations#
  mutate(Location = tolower(Location),  #turn to lowercase#
       Location = str_split(Location, ";")) %>%  #split at the ;#
  unnest(Location) %>% #unnest the vector#
  mutate(Evidence = str_extract_all(Location, "eco:\\d+"), #create a new column with the evidence#
       Evidence = map(Evidence, unique),  #
       Location = str_replace_all(Location, " \\{.*\\}", ""))  %>% #erase the string inside {}#
  mutate(SCpathway = grepl(paste(c("lysosome","endoplasmic reticulum","golgi","vesicle"),collapse = "|"), Loca-
tion)) %>%
  group_by(Entry) %>%
  mutate(SCpathway = any(SCpathway == TRUE)) %>%
  select(-Location,-Evidence) %>%
  distinct(.keep_all = TRUE)
#in general secretory pathway proteins
df5 <- read_tsv("uniprot-all.tab.gz") %>% ##human reviewed##
  rename(Location = `Subcellular location [CC]`) %>%
  mutate(Location = str_replace_all(Location, "SUBCELLULAR LOCATION: ", ""), #erase the SUBCELLULAR string#
       Location = str_replace_all(Location, "Note=.*$", ""),  ##erase the Note:... till end#
       Location = str_replace_all(Location, "Ref\\.", "Ref"),
```

```
      Location = str_split(Location, "\\."),
      Location = map(Location, str_trim)) %>% #trim whitespace from start and end#
  unnest(Location) %>%   #unnest the vector of location#
  filter(!is.na(Location), Location != "") %>%   #remove the NA locations and blank locations#
  mutate(Location = tolower(Location),  #turn to lowercase#
      Location = str_split(Location, ";")) %>%  #split at the ;#
  unnest(Location) %>% #unnest the vector#
  mutate(Evidence = str_extract_all(Location, "eco:\\d+"), #create a new column with the evidence#
      Evidence = map(Evidence, unique),  #
      Location = str_replace_all(Location, " \\{.*\\}", ""))  %>% #erase the string inside {}#
  mutate(SCpathway = grepl(paste(c("lysosome","endoplasmic reticulum","golgi","vesicle"),collapse = "|"), Loca-
tion)) %>%
  group_by(Entry) %>%
  mutate(SCpathway = any(SCpathway == TRUE)) %>%
  select(-Location,-Evidence) %>%
  distinct(.keep_all = TRUE)


pos <- which(df5$Entry %in% intersect(df5$Entry,df2$Entry))
df5 <- df5[pos,]
df5 <- full_join(df5,df2)
FPsSCpath <- df4[df4$SCpathway == TRUE,]
SCpathproteins <- df5[df5$SCpathway == TRUE & df5$hasSP == 1 & df5$Location != "Secreted",]
pos2 <- which(SCpathproteins$Entry %in% intersect(SCpathproteins$Entry,df4$Entry))

#false positives in the secretory pathway SP=1 not secreted
FPsSCpath <- SCpathproteins[pos2,]
#correctly classified as not secreted but in the Sc pathway SP=1
CorrectSCpath <- SCpathproteins[-pos2,]

#why were those correctly classified but the others were not can i explain that with the features that i have???
pos3 <- which(play2$Entry %in% intersect(play2$Entry,CorrectSCpath$Entry))
corcls <- play2[pos3,]
pos4 <- which(play2$Entry %in% intersect(play2$Entry,FPsSCpath$Entry))
incorcls <- play2[pos4,]
```